Efficient estimation of probability of conflict between air traffic using Subset Simulation

Chinmaya Mishra, Simon Maskell, Siu-Kui Au and Jason F. Ralph
{cmishra, smaskell, siukuiau, jfralph}@liverpool.ac.uk
Department of Electrical Engineering and Electronics, University of Liverpool, United Kingdom

***Abstract*—This paper presents an efficient method for estimating the probability of conflict between air traffic within a block of airspace. Autonomous Sense-and-Avoid is an essential safety feature to enable Unmanned Air Systems to operate alongside other (manned or unmanned) air traffic. The ability to estimate probability of conflict between traffic is an essential part of Sense-and-Avoid. Such probabilities are typically very low. Evaluating low probabilities using naive Direct Monte Carlo generates a significant computational load. This paper applies a technique called Subset Simulation. The small failure probabilities are computed as a product of larger conditional failure probabilities, reducing the computational load whilst improving the accuracy of the probability estimates. The reduction in the number of samples required can be one or more orders of magnitude. The utility of the approach is demonstrated by modeling a series of conflicting and potentially conflicting scenarios based on the standard Rules of the Air.**

***Index Terms*—Air traffic, Probability of conflict, Subset Simulation, Direct Monte Carlo, Metropolis Hastings, Sense-and-Avoid, Benchmarking**

I. Introduction

FUTURE autonomous operations of Unmanned Air Systems (UAS) within densely populated airspace require an automated Sense-and-Avoid (SAA) system [1]. A key element within the Sense-and-Avoid (SAA) topic is Conflict Detection and Resolution (CD&R) [1]. A conflict occurs when the separation between any aircraft or obstacle reduces below a minimum distance. Such a situation could − in the worst case − generate a collision between air vehicles but even in the absence of an actual collision it will violate the mandated Rules of the Air, and may give rise to an air incident. Such incidents must be reported as soon as possible to the local Air Traffic Service Unit (ATSU) [2].

Initial work on CD&R can be found in robotics where the collision avoidance problem has been treated as a path planning task [3] and an early approach to the collision avoidance problem involved using artificial potential fields [4]. Such methods are suitable for scenarios where movement of the vehicles may be relatively slow, restricted in space or in scope. However, over the following decades the increased use of UAS has created demand for autonomous CD&R solutions which are suitable for the more dynamic aerospace environment. A large number of CD&R methods have been proposed during this period and comprehensive surveys have been conducted by Kuchar and Yang [5], Krozel et al. [6], Warren [7] and Zeghal [8]. Kuchar and Yang have proposed a taxonomy of methods useful in identifying gaps and directing future efforts within the SAA community [5]. More recently, Albaker and Rahim have presented an up to date survey of CD&R methods for UAS [9]. The work presented in this paper can be categorized as a Conflict Detection method that can be used for both cooperative and non-cooperative scenarios.

The CD&R methods are broadly categorized as cooperative and non-cooperative. Cooperative methods assume that traffic shares relevant information via radio, data link or by contacting ground based ATSU. These methods are dependent on cooperative equipment such as Transponders and/or Automatic Dependent Surveillance-Broadcast (ADS-B) that are carried on-board the aircraft. This equipment declares the current state of the aircraft to nearby traffic. If the potential for a conflict is identified the situation will be resolved by coordinating maneuvers between the traffic, often via two-way radio communications. The maneuvers are dictated by following a set of customary rules that determine the right-of-way for each aircraft. These are based on existing Visual Flight Rules (VFR) within the civil aviation domain [10]. In VFR, it is the flight crew's responsibility to maintain safe separation with traffic. In the absence of visual information (due to limited visibility caused by bad weather), the flight crew must rely on external information. In such situations, Instrument Flight Rules (IFR) are used with the ATSU monitoring traffic separation using Radar and then directing the flight crew so as to maintain safe separation. Alternatively, on larger aircraft, a Traffic Alert Collision Avoidance System (TCAS) [11] can be used. The TCAS system provides Resolution Advisories (RA) to flight crews of conflicting traffic in the form of maneuvers to be followed to resolve the conflict. In each case, a potential conflict is resolved in accordance with the rules given by the local aviation authority for the airspace within which the aircraft are operating; such as the Federal Aviation Administration (FAA) in the US [12] or the Civil Aviation Authority (CAA) in the UK [13]. The rules stated by most aviation authorities are based on the rules outlined by the International Civil Aviation Organization (ICAO) [14]. When a conflict type is identified the appropriate resolution maneuver is executed. For example, when aircraft are approaching each other head-on the rules will say that both aircraft maneuver to their right. All traffic involved with the conflict must cooperate for a successful resolution [15]. Each of these methods assumes that all aircraft involved in the potential conflict are sharing information and behaving in accordance with the accepted Rules of the Air [10].

In contrast, non-cooperative methods assume that no infor-

mation related to the current state or future intent of traffic has been shared (i.e. there is no flight plan exchange or radio/data link). This is a far more challenging problem since information related to traffic state and intentions must be measured or inferred from the behavior of non-cooperative aircraft. Normally, this will be due to the lack of appropriate technology on-board the aircraft: for example, a lightweight commercial of-the-shelf (COTS) UAS, obtained by the general public and used for recreational purposes. Problems occur when these aircraft are operated within non-segregated airspace. This type of airspace contains aircraft (manned or unmanned) that adhere to the Rules of the Air and expect traffic to do so as well. The lack of cooperative technology on-board a lightweight UAS prevents awareness of traffic and increases the risk of a midair collision. This problem needs to be addressed due to the increased number of near miss incidents involving such UAS operating within non-segregated airspace [16]. The problem of the lack of information is addressed by using on-board sensors. Information related to state of traffic is obtained from observations using sensors such as Radar, Lidar and/or cameras. For example, Mcfadyen et al. have considered using visual predictive control with a spherical camera model to create a collision avoidance controller [17]. Recently, Huh et al. have proposed a vision based Sense-and-Avoid framework that utilizes a camera to detect and avoid approaching airborne intruders [18]. A collision avoidance system that uses a combination of Radar and electro-optical sensors have been prototyped and tested by Accardo et al [19]. Measurement data obtained from sensors are inherently noisy. This gives rise to uncertainties in the observed state and predicted motion of the non-cooperative aircraft. In an environment where future trajectories are uncertain, the likelihood of a conflict is an essential metric. Obtaining an accurate estimate for the Probability of Conflict (P_c), given the sensor data, is a key parameter required to resolve traffic conflicts. This paper provides a method to calculate the P_c metric that is more efficient than the standard approach of using Direct Monte Carlo (DMC) methods.

Probabilistic methods for conflict resolution requiring the calculation of metrics like the Probability of Conflict (P_c) have been discussed in [5]. Nordlund and Gustafsson [20] noted the huge number of simulations required to get sufficient reliability for small risks and suggested an approach that reduced the three dimensional problem to a one dimensional integral along piecewise straight paths [21], [22]. More recently, Jilkov et al. have extended a method developed by Blom and Bakker [23] and estimated P_c using multiple models for aircraft trajectory prediction [24]. Many probabilistic methods involve the use of Monte Carlo methods where uncertainties exist and Monte Carlo methods can be found in existing CD&R methods [24]–[31]. Unfortunately, for scenarios where the expected P_c is low, a Monte Carlo method will require a very large number of simulations to estimate P_c with any accuracy. To reduce the computational cost associated with Monte Carlo methods, Prandini et al. have estimated the risk of conflict using the Interacting Particle System (IPS) method [32]. This method fixes a set of initial conditions of the aircraft and alters reducing subsets of the propagated trajectories to satisfy the intermediate thresholds; this assumes that the predicted trajectories are non-deterministic with the probability of conflict being associated with outliers in the propagation, not outliers in the initial conditions. If, however, the trajectory is deterministic (or near-deterministic), then IPS is unable to provide improved computational efficiency relative to direct (Monte Carlo) sampling. This paper proposes the use of the Subset Simulation method [33] to avoid this problem and allows the initial conditions to be adjusted as the subsets are navigated. Subset Simulation approaches the problem of reducing the computational load associated with calculating low probabilities by focusing the simulation towards the rare regions of interest within the probability distribution function (pdf). The regions of interest correspond to the events which may lead to conflict between traffic.

Originally, Au and Beck proposed Subset Simulation as a method for computing small failure probabilities as a result of (larger) conditional failure probabilities [33]. The method was proposed in Civil Engineering to compute probabilities of structural failure and identify associated failure scenarios [34]. The focus of their work was on understanding the risk to structures posed by seismic activity. An example of application of Subset Simulation in the Aerospace discipline was the efficient estimation of the amount of propellant required by a spacecraft to perform attitude control [35]. The accuracy of the probabilities estimated using Subset Simulation have been compared with the accuracy of probabilities estimated using Direct Monte Carlo across a set of benchmarks shown in [36]. The results show that the accuracy of the estimates for low probabilities using Subset Simulation is better than the accuracy of estimates for low probabilities using Direct Monte Carlo. This paper modifies the methods developed by Au and Beck [37] and demonstrates that they can significantly reduce the computational load required to estimate the value of P_c for air traffic within a block of airspace by reducing the number of samples required. The proposed method is applied to a set of conflicting and potentially conflicting test scenarios based on the Rules of the Air specified by aviation authorities. Since these scenarios are standard engagements considered by aviation authorities, they could also be used as a benchmark for comparison against future methods. The P_c during some scenarios is low; despite this, it is essential to provide an approximation this metric due to the catastrophic nature of a collision.

The paper is structured as follows: sections II and III describe the Direct Monte Carlo (DMC) and Metropolis Hastings (MH) methods respectively. The Subset Simulation theory is based on a combination of DMC and MH methods. Section IV describes Subset Simulation. Section V then describes a method to use Subset Simulation to estimate the probability of conflict between air traffic for scenarios described by the Right-of-way rules specified by the ICAO [10]. Section VI presents simulation results of estimating P_c between air traffic for conflicting and potentially conflicting scenarios. Section VII analyzes the efficiency and accuracy of estimating the P_c using Subset Simulation and Direct Monte Carlo. Finally, section VIII concludes the paper.

Algorithm 1 Determine distance between samples X and C

function H(X,C)
  $V = X - C$
  $R = \sqrt{V_x^2 + V_y^2}$
  return R
end function

Algorithm 2 Direct Monte Carlo

function DMC(N, C, r_c)
  $D = 0$
  for $n = 1 : N$ **do**
    $x \sim \mathcal{N}(0, 1)$
    $y \sim \mathcal{N}(0, 1)$
    $X_n = [x, y]^{\mathrm{T}}$
    $R_n = \mathrm{H}(X_n, C)$
    if $R_n \leq r_c$ **then**
      $D = D + 1$
    end if
  end for
  $P_F = \frac{D}{N}$
  return P_F
end function

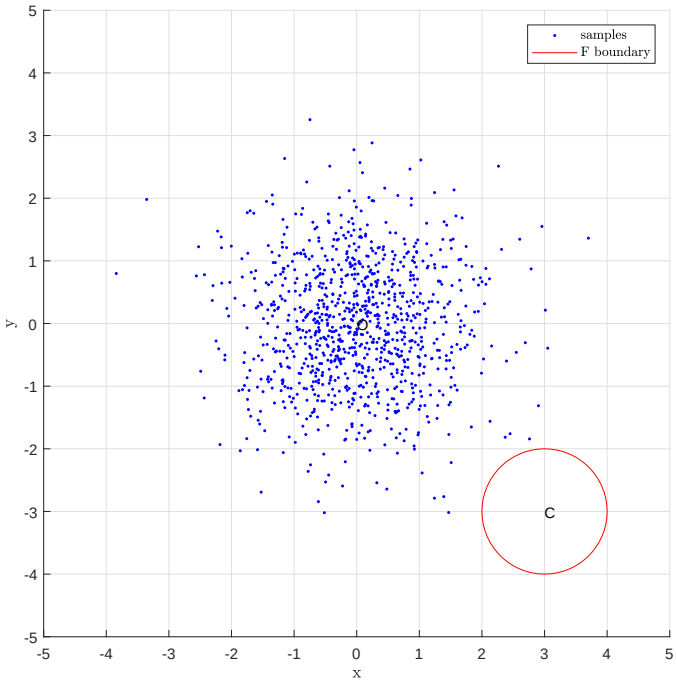
II. Direct Monte Carlo

The Direct Monte Carlo (DMC) method is a sampling method that can be used to characterize a distribution of interest. The objective of this section is to estimate the probability of a type of event to occur. Therefore the DMC method is used as a 'statistical averaging' tool, where the probability of failure P_F is estimated as the ratio of failure responses to the total number of trials [37].

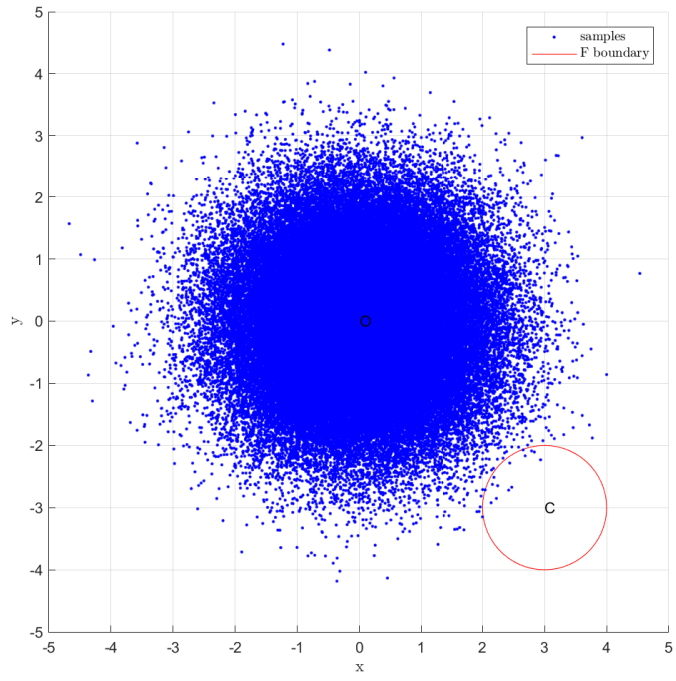
A set of N independent identically distributed (i.i.d) inputs $\{X_n : n = 1, ..., N\}$ are drawn from the proposal distribution $q(X|\mu, \sigma^2)$ of the input parameter space. The proposal distribution can be any known distribution that can be sampled from. We choose a Normal distribution that is centered at the mean μ and has a variance of σ^2. A set of system responses are observed $\{Y_n = h(X_n) : n = 1, ..., N\}$, where $h(...)$ is the system process. The occurrence of a failure event F is indicated when a scalar quantity b_F (threshold) is exceeded. The number of samples that exceed the threshold is Y_F. Therefore the probability of failure is estimated as $P_F = P(Y \geq b_F) = \frac{Y_F}{N}$. Such an approach is suitable for large probabilities (such as $P > 0.1$) where a small number of samples can be used to estimate the probability. However for small probabilities (such as the tail region of the pdf, where $P \leq 10^{-3}$) a significantly large number of samples must be drawn to accurately estimate the probability. This is illustrated by the following example.

A. Estimating probability of drawing samples from region F

Fig. 1 shows a 10×10 square centered at $O = [0, 0]^{\mathrm{T}}$. The region F is a circle with radius $r_c = 1$, centered at $C = [3, -3]^{\mathrm{T}}$ within this square. The objective is to estimate the probability of drawing samples from this region. The probability distribution of the overall area is represented by a Gaussian distribution centered at $O = [0, 0]^{\mathrm{T}}$. A set of N samples $\{X_n : n = 1, ..., N\}$ are drawn where each sample is a vector; $X_n = [x_n, y_n]^{\mathrm{T}}$. The x and y values of each sample are the x-coordinate and y-coordinates of the position respectively. To clarify, $X_1 = [x_1, x_2]^{\mathrm{T}}$ where $x_1 \sim \mathcal{N}(0, 1)$ and $y_1 \sim \mathcal{N}(0, 1)$. The distance between the position of each sample and center of circle C is $\{R_n = H(X_n, C) : n = 1, ...N\}$ as defined by Algorithm 1. To clarify, the distance



(a) Direct Monte Carlo with 100 samples



(b) Direct Monte Carlo with 10^5 samples

Fig. 1. The probability of drawing samples from the region F is estimated using Direct Monte Carlo. Fig. 1(a) estimates the $P_F = 0$ with 100 samples. Fig. 1(b) estimates the $P_F = 1.5 \times 10^{-4}$ with 10^5 samples.

between sample X_1 and C is $R_1 = H(X_1, C)$. Algorithm 2 is used to estimate the probability of drawing samples from the region F.

Fig. 1(a) shows 100 samples drawn from the distribution. Note, no samples are drawn from the area F. The probability is estimated $P_F = 0$. The number of samples are increased to $N = 10^5$. Fig. 1(b) shows some samples are drawn from the region F and the probability is estimated $P_F = 1.5 \times 10^{-4}$. This illustrates that Direct Monte Carlo requires a significantly large number of samples to estimate the probability of drawing samples from the region F.

This method estimates P_F by attempting to realize the entire pdf centered at O that includes the area F. As the area F reduces (to represent estimating lower target probabilities) the number of samples required to estimate P_F increases making such an approach computationally demanding. A different algorithm is needed.

III. Metropolis Hastings

Metropolis-Hastings (MH) is a Markov Chain Monte Carlo (MCMC) method used to characterize a distribution of interest by sampling from a known distribution. We refer to this distribution of interest as the target distribution. The MH algorithm originates from the Metropolis algorithm first used in statistical Physics by Metropolis and co-workers (Metropolis et al, 1953) [38]. Hastings proposed a generalized form of this algorithm leading to the Metropolis Hastings (MH) algorithm [39].

The MH method generates samples from the proposal distribution $q(X|x_0, \sigma^2)$ by starting from a seed value x_0. A chain of n samples is then generated, starting with x_0. The sample x_{k+1} is generated from the current sample x_k using the following steps [37]:

1) Generate a candidate sample $x^* \sim q(x^*|x_k, \sigma^2)$.
2) Calculate an acceptance ratio: $\alpha = \frac{q(x_k|x^*,\sigma^2)f(x^*)}{q(x^*|x_k,\sigma^2)f(x_k)}$
3) Draw a sample e from a uniform distribution [0,1]
4) Set $x_{k+1} = \begin{cases} x^* & \text{if } e < \alpha \\ x_k & \text{otherwise} \end{cases}$
5) Repeat steps 1 to 4 until n samples have been generated.

The function $f(...)$ defines the target density for the input sample. While, $n \to \infty$, this process is guaranteed to accept samples from q that leads to the realization of the target distribution [40]. To help ensure that all regions of the target density are explored, multiple seeds can be used to generate multiple chains of samples in parallel [37].

A. Drawing samples from the region F

The Metropolis Hastings method is defined in algorithm 3 and it is applied to the example of estimating the probability of drawing samples from region F as shown in the previous section. The covariance of the proposal σ^2 is a 2×2 identity matrix $I_{2\times2}$ and the covariance of the distribution of interest $\sigma^2_{r_c} = r_c^2 \times I_{2\times2}$ where r_c is the radius of the region F. For this example $r_c = 1$, therefore $\sigma^2_{r_c} = I_{2\times2}$.

Fig. 2 illustrates the chains of samples generated by the Metropolis Hastings algorithm. This figure shows 10 samples drawn from the proposal distribution using the DMC method.

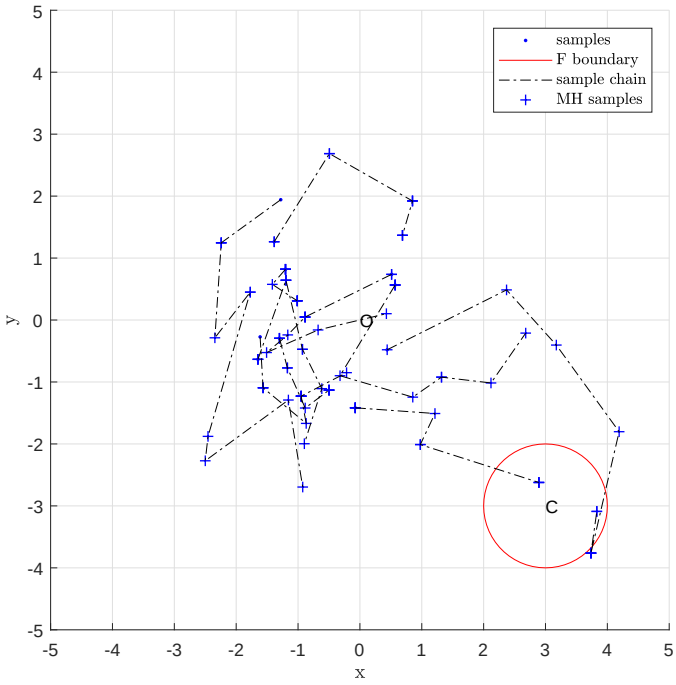


Fig. 2. Drawing samples from the region F using Metropolis Hastings algorithm to generate chains of conditional samples. The initial samples used as seeds are drawn using Direct Monte Carlo.

Algorithm 3 Generate conditional chains of samples using Metropolis Hastings algorithm

function MH(s, n, C, r_c)
  $\sigma^2_{r_c} = r_c^2 I_{2\times2}$
  for $j = 1 : |s|$ **do** ▷ *For each seed*
    $X_0 = s_j$ ▷*Select seed sample*
    for $k = 0 : n-1$ **do**
    ▷*Generate Candidate sample* X^*
      $g \sim \mathcal{N}(0, 1)$
      $X^* = X_k + g$
      ▷*Calculate acceptance ratio*
      $\beta = \frac{q(X^*|X_k,\sigma^2)}{q(X_k|X^*,\sigma^2)} \frac{p(X^*|C,\sigma^2_{r_c})}{p(X_k|C,\sigma^2_{r_c})}$
      $\alpha = min\{1, \beta\}$
      $e \sim [0, 1]$
      $X^{(j)}_{k+1} = \begin{cases} X^* & \text{if } e < \alpha \\ X_k & \text{if } e \geq \alpha \end{cases}$
    end for
  end for
  return $X^{(j)}$
end function

These samples are seeds $s = \{X_1, ..., X_{10}\}$. The MH algorithm is applied using the seeds s. Each seed generates a chain of 10 samples. Note that many sample chains do not reach the region F. It is clear that it might be more efficient to generate more samples for chains with seeds that are closer to the region F since they have higher likelihood of generating samples that are within the region F or closer to the region F. Subset Simulation achieves this and is described in the next section.

Level 0			Level i				Level $m-1$		
$\mathbf{P}_n^{(0)}$	$\mathbf{B}_n^{(0)}$	$\tilde{\mathbf{X}}_n^{(0)}$	$\mathbf{P}_n^{(i)}$	$\mathbf{B}_n^{(i)}$	$\tilde{\mathbf{X}}_n^{(i)}$	.....	$\mathbf{P}_n^{(m-1)}$	$\mathbf{B}_n^{(m-1)}$	$\tilde{\mathbf{X}}_n^{(m-1)}$
$P_1^{(0)}$	$B_1^{(0)}$	$\tilde{X}_1^{(0)}$							
$\vdots$	$\vdots$	$\vdots$							
$P_{N-N_c}^{(0)}$	$B_{N-N_c}^{(0)}$	$\tilde{X}_{N-N_c}^{(0)}$							
$P_{N-N_c+1}^{(0)}$	$B_{N-N_c+1}^{(0)}$	$\tilde{X}_{N-N_c+1}^{(0)}$	$P_1^{(i)}$	$B_1^{(i)}$	$\tilde{X}_1^{(i)}$				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$				
$P_N^{(0)}$	$B_N^{(0)}$	$\tilde{X}_N^{(0)}$	$P_{N-N_c}^{(i)}$	$B_{N-N_c}^{(i)}$	$\tilde{X}_{N-N_c}^{(i)}$				
			$P_{N-N_c+1}^{(i)}$	$B_{N-N_c+1}^{(i)}$	$\tilde{X}_{N-N_c+1}^{(i)}$	.....	$P_1^{(m-1)}$	$B_1^{(m-1)}$	$\tilde{X}_1^{(m-1)}$
			$\vdots$	$\vdots$	$\vdots$	.....	$\vdots$	$\vdots$	$\vdots$
			$P_N^{(i)}$	$B_N^{(i)}$	$\tilde{X}_N^{(i)}$	.....	$P_{N-N_c}^{(m-1)}$	$B_{N-N_c}^{(m-1)}$	$\tilde{X}_{N-N_c}^{(m-1)}$
						.....	$P_{N-N_c+1}^{(m-1)}$	$B_{N-N_c+1}^{(m-1)}$	$\tilde{X}_{N-N_c+1}^{(m-1)}$
							$\vdots$	$\vdots$	$\vdots$
							$P_N^{(m-1)}$	$B_N^{(m-1)}$	$\tilde{X}_N^{(m-1)}$

Fig. 3. The table illustrates a general form of Subset Simulation. The probability intervals $\mathbf{P}_n^{(0)}$, quantity of interest $\mathbf{B}_n^{(0)}$ and respective samples $\tilde{\mathbf{X}}_n^{(0)}$ are generated at each level of Subset Simulation. The samples $\tilde{X}_{N-N_c+1}^{(0)}$ to $\tilde{X}_N^{(0)}$ generated in level 0 are used as seeds to generate conditional samples using MH in the subsequent level of subset simulation. This process continues until the final level $(m-1)$ of subset simulation is reached.

IV. Subset Simulation

Subset Simulation (SS) is based on a combination of Direct Monte Carlo (DMC) and Metropolis Hastings (MH) methods as described in sections II and III respectively. It calculates the probability of rare events occurring as the product of the probabilities of less-rare events. Such an approach is less computationally expensive than either DMC or MH alone. A general outline of the SS method is presented in this paper and the interested reader is referred to [37] for more details.

Subset Simulation generates a Complimentary Cumulative Distribution Function (CCDF) of the response quantity of interest Y. The probability of failure P_F can be directly estimated from the CCDF. This CCDF is constructed by generating samples that satisfy a series of intermediate thresholds $b_1 > b_2 > b_3 > ... > b_{m-1}$ that divide the space into m nested regions. These thresholds are adaptively defined as the simulation progresses. This is described later on in this section. The threshold b_{m-1} is the required failure threshold b_F ($b_{m-1} = b_F$). The intermediate thresholds allow the probability of failure to be estimated using a classical conditional structure given by

$$P_F = P(Y < b_{m-1}|Y < b_{m-2})P(Y < b_{m-2}) \qquad (1)$$

Samples are generated to satisfy the threshold for each level. The total number of levels m is dependent on the magnitude of the target probability P_F. Subset Simulation uses 'level probability' $p_0 \in (0, 1)$ to control how quickly the simulation reaches the target event of interest [37]. The target probability is used to approximate the number of levels m required by evaluating $P_F = (p_0)^m$. To clarify, if the target probability is $P_F = 10^{-5}$ and $p_0 = 0.1$ then the total number of levels required will be $m = 5$.

A. Level 0

Subset Simulation begins at level $i = 0$ with Direct Monte Carlo (DMC) sampling from the entire region of interest. A set of N samples $\{X_n^{(0)} : n = 1, ..., N\}$ are drawn from a proposal distribution $q(X_n^{(0)}|\mu, \sigma^2)$ (as described in section II). The set of output responses $Y_n^{(0)}$ are evaluated $\{Y_n^{(0)} = h(X_n^{(0)}) : n = 1, ..., N\}$. The function $h(...)$ defines the system response to the input sample. In the context of SS, the responses $Y_n^{(0)}$ are also known as the quantity of interest. The set $Y_n^{(0)}$ is sorted in descending order to create the set $\{B_n^{(0)} : n = 1, ..., N\}$. The input samples $X_n^{(0)}$ are reordered $\tilde{X}_n^{(0)}$ and correspond to the sorted quantity of interest $B_n^{(0)}$. To clarify, $\tilde{X}_1^{(0)}$ is the input sample that generates the largest output $B_1^{(0)}$. A CCDF is generated by plotting $B_n^{(0)}$ against the probability intervals $P_n^{(0)}$. The probability intervals $P_n^{(0)}$ are generated using the following equation:

$$P_n^{(i)} = p_0^i \frac{N-n}{N} \quad n = 1.....N \qquad (2)$$

The vector of probability intervals $P_n^{(0)}$ is concatenated with the sorted quantity of interest $B_n^{(0)}$ and their respective samples $\tilde{X}_n^{(0)}$ as illustrated in the table shown in Fig. 3 by the column titled 'Level 0'.

The set of probability intervals $P_n^{(0)}$ are plotted against $B_n^{(0)}$ to generate the CCDF. Level 0 makes it possible to accurately approximate CCDF values from $1 - N^{-1}$ to p_0. Typically the region of interest within the pdf is outside this range (since SS is typically used to realize rare events). To explore probabilities below p_0, further levels of simulation must be conducted.

B. Level $i > 0$

The subsequent levels of SS where, $i > 0$ explore the rarer regions of the probability distribution. This is achieved by

$\mathbf{P}_n$	$\mathbf{B}_n$	$\tilde{\mathbf{X}}_n$	
$P_1^{(0)}$	$B_1^{(0)}$	$\tilde{X}_1^{(0)}$	
$\vdots$	$\vdots$	$\vdots$	*Level 0*
$P_{N-N_c}^{(0)}$	$B_{N-N_c}^{(0)}$	$\tilde{X}_{N-N_c}^{(0)}$	*samples retained*
$P_1^{(i)}$	$B_1^{(i)}$	$\tilde{X}_1^{(i)}$	
$\vdots$	$\vdots$	$\vdots$	*Level i*
$P_{N-N_c}^{(i)}$	$B_{N-N_c}^{(i)}$	$\tilde{X}_{N-N_c}^{(i)}$	*samples retained*
$\vdots$	$\vdots$	$\vdots$	
$P_1^{(m-1)}$	$B_1^{(m-1)}$	$\tilde{X}_1^{(m-1)}$	
$\vdots$	$\vdots$	$\vdots$	
$P_{N-N_c}^{(m-1)}$	$B_{N-N_c}^{(m-1)}$	$\tilde{X}_{N-N_c}^{(m-1)}$	
$P_{N-N_c+1}^{(m-1)}$	$B_{N-N_c+1}^{(m-1)}$	$\tilde{X}_{N-N_c+1}^{(m-1)}$	
$\vdots$	$\vdots$	$\vdots$	*Level m − 1*
$P_N^{(m-1)}$	$B_N^{(m-1)}$	$\tilde{X}_N^{(m-1)}$	*samples retained*

Fig. 4. This table illustrates the concatenation of probability intervals samples, quantity of interest and samples generated at each level of Subset Simulation. Note the samples used as seeds from the previous level are discarded and replaced with the samples generated at the current level.

generating multiple chains of conditional samples using the MH method as discussed in the previous section. The number of chains and number of samples per chain are N_c and N_s respectively. They are determined as

$$N_c = p_0 N \tag{3}$$

$$N_s = p_0^{-1} \tag{4}$$

Each level of Subset Simulation maintains N samples ($N = N_c N_s$). The response values of conditional samples generated for the current level i must not exceed the intermediate threshold b_i for this level. This threshold is determined by

$$b_i = B_{N-N_c}^{(i-1)} \quad i \text{ is the current subset level} \tag{5}$$

The intermediate threshold for level $i = 1$ is $b_1 = B_{N-N_c}^{(0)}$. To clarify the intermediate threshold is the $(N - N_c)^{\text{th}}$ element of the sorted set of response values $B_n^{(0)}$. The set of seeds $s_j^{(i)}$ are used to generate samples for the current level i are samples generated from the previous level $(i-1)$ are defined by

$$s_j^{(i)} = \tilde{X}_n^{(i-1)} \tag{6}$$

where $1 \leq j \leq N_c$, $(N - N_c + 1) \leq n \leq N$ and $i > 0$.

The set of seeds used to generate conditional samples for level $i = 1$ is $s^{(1)} = \{\tilde{X}_{N-N_c+1}^{(0)}, ..., \tilde{X}_N^{(0)}\}$. The N conditional samples $X_n^{(1)}$ are generated using the MH method. The quantities of interest for $X_n^{(1)}$ are determined $\{Y_n^{(1)} = h(X_n^{(1)}) : n = 1, ..., N\}$ and are sorted in the same manner as the previous level $B_n^{(1)}$. The set $B_n^{(1)}$ and respective samples $\tilde{X}_n^{(1)}$ are concatenated with the probability intervals $P_n^{(1)}$ as illustrated in the table shown in Fig. 3 by the column titled 'Level i'. Note the samples $\{\tilde{X}_{N-N_c+1}^{(0)}, ..., \tilde{X}_N^{(0)}\}$ shown in the column titled 'Level 0' are used as seeds to generate the conditional samples $\{\tilde{X}_1^{(i)}, ..., \tilde{X}_N^{(i)}\}$ in column titled 'Level i'.

This process is continued until the target level of probability $(p_0)^m$ is reached at level $i = m - 1$; as shown by the column titled 'Level $m - 1$'. The samples used as seeds to generate samples for the consecutive level are discarded and replaced with the generated samples. This is illustrated in the table shown in Fig. 4. The column of probability intervals $\mathbf{P}_n$ is plotted against the respective quantities of interest $\mathbf{B}_n$ to generate a CCDF.

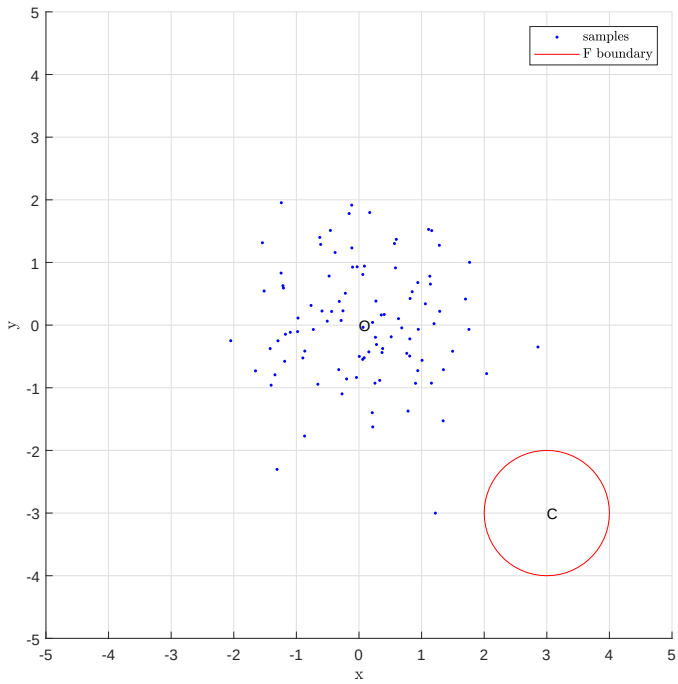
This method is continued until the target level of probability $P_F = (p_0)^m$ is reached. By generating and evaluating conditional samples, the output samples tend towards the target distribution with significantly less trials than are needed when using the DMC method. The progressive nature of the algorithm can be demonstrated in the example problem of estimating the probability of drawing samples from the region F.

Algorithm 4 Generate conditional chains of samples of Subset Simulation using Metropolis Hastings algorithm

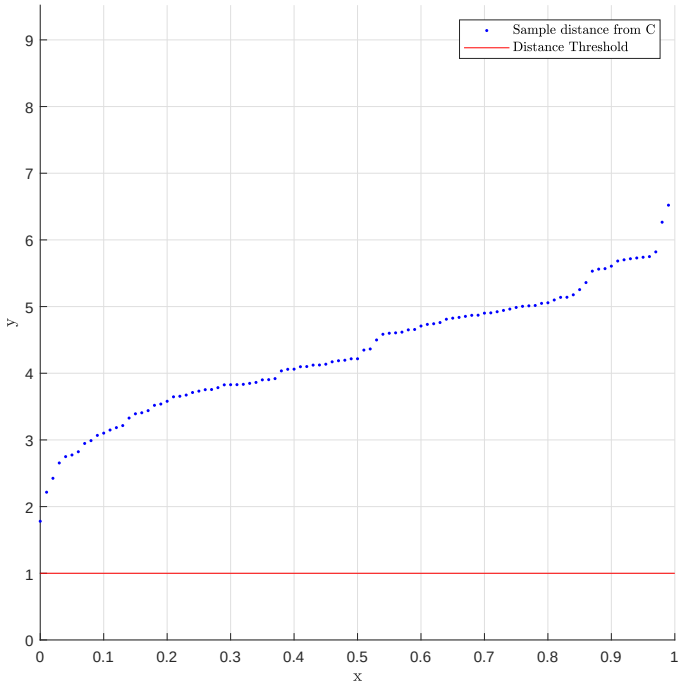
function MH_I(s, n, C, r_c)
  $\sigma_{r_c}^2 = r_c^2 I_{2\times 2}$
  for $j = 1 : |s|$ **do** ▷ *For each seed*
    $X_0 = s_j$ ▷*Select seed sample*
    for $k = 0 : n - 1$ **do**
    ▷*Generate Candidate sample* X^*
      $g \sim \mathcal{N}(0, 1)$
      $X^* = X_k + g$
      ▷*Determine distance between* X^* *and C*
      $R^* = \mathrm{H}(X^*, C)$
      ▷*Determine distance between* X_k *and C*
      $R_k = \mathrm{H}(X_k, C)$
      ▷*Indicator function for range*
      $d = \begin{cases} 1 & \text{if } R^* \leq r_c \\ 0 & \text{if } R^* > r_c \end{cases}$
      ▷*Calculate acceptance ratio*
      $\beta = \frac{q(X^*|X_k,\sigma^2)}{q(X_k|X^*,\sigma^2)} \frac{p(X^*|C,\sigma_{r_c}^2)}{p(X_k|C,\sigma_{r_c}^2)}$
      $\alpha = min\{1, \beta\}$
      $e \sim [0, 1]$
      $X_{k+1}^{(j)} = \begin{cases} X^* & \text{if } e < \alpha \\ X_k & \text{if } e \geq \alpha \end{cases}$
      $R_{k+1}^{(j)} = \begin{cases} R^* & \text{if } e < \alpha \\ R_k & \text{if } e \geq \alpha \end{cases}$
    end for
  end for
  return $X^{(j)}, R^{(j)}$
end function

C. Estimating Probability of drawing samples from region F

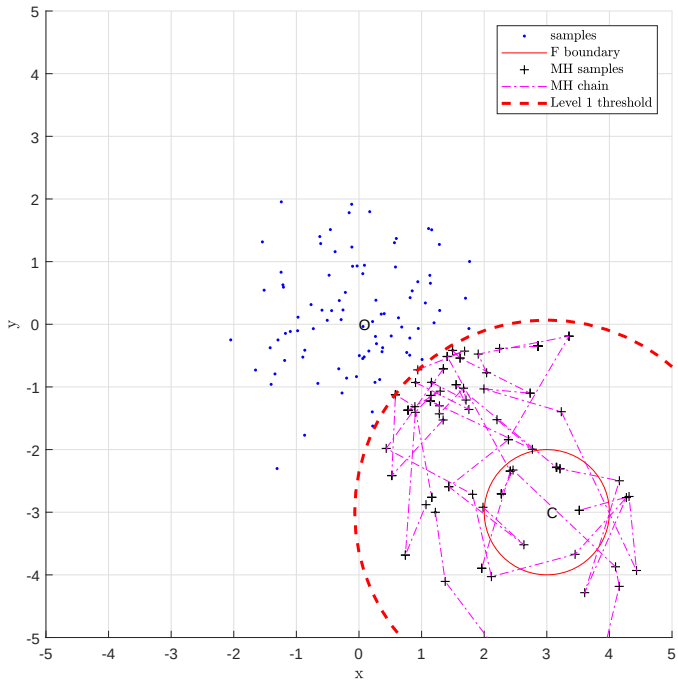
The example of estimating the probability of drawing samples from the region F shown in the previous sections is used to illustrate the Subset Simulation method (using algorithm 5).



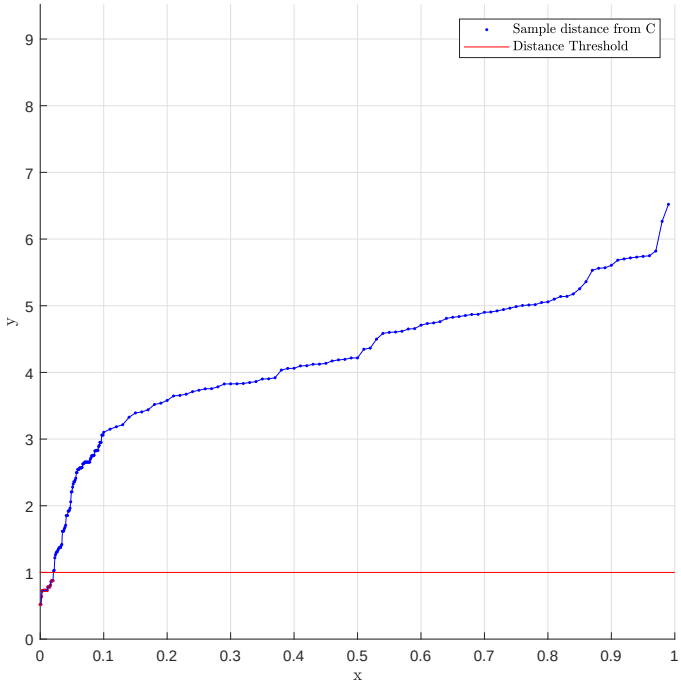
(a) Subset Simulation Level 0



(b) Subset Simulation Level 0 CCDF



(c) Subset Simulation Level 1



(d) Subset Simulation Level 1 CCDF

Fig. 5. Subset Simulation is applied to the problem of estimating the probability of drawing samples from the region F. Subset Simulation begins with level 0 by drawing N = 100 samples from a Gaussian distribution centered at $O = [0, 0]$ using the DMC method as shown in Fig. 5(a). The quantity of interest is the distance between each sample and C. These are plotted against probability intervals to generate a CCDF as shown in Fig. 5(b). No samples are within the region F. The SS method proceeds to level 1 and conditional samples are generated using the MH method. The N_c level 0 samples are used to generate the conditional samples shown in Fig. 5(c). These conditional samples are drawn progressively closer to the region F until some samples are drawn from the region F. This is achieved by drawing samples from intermediate thresholds closer to the boundary of F. The quantity of interest for the samples are determined and plotted against the probability intervals for the current level. This CCDF is appended to the previous CCDF by replacing the samples used as seeds from the previous level as shown in Fig. 5(d).

The radius of the circle bounding the region F is $r_c = 1$. The SS parameters used for this example are: $p_0 = 0.1$, $N = 100$, $N_s = 10$, $N_c = 10$, $m = 2$. Subset Simulation is typically used to realize rare events (for $P_F \leq 10^{-3}$ therefore $m > 3$). However for the purpose of this example the number of levels is kept low ($m = 2$).

Algorithm 5 Subset Simulation

function SS(C, N, p_0, m)
  $N_c = p_0 N$
  $N_s = p_0^{-1}$
  $i = 0$ *Set current level*
  ▷*Direct Monte Carlo: Draw N samples and determine quantity of interest*
  for $n = 1 : N$ **do**
    $X_n^{(i)} \sim \mathcal{N}(0, 1)$
    ▷*Quantity of interest: Determine distance between samples* $X_n^{(i)}$ *and* C
    $R_n^{(i)} = \mathrm{H}(X_n^{(i)}, C)$
  end for
  $B_n^{(i)} \leftarrow R_n^{(i)}$ *Sort distances in descending order*
  $\tilde{X}_n^{(i)} \leftarrow X_n^{(i)}$ *Reorder the input samples to correspond to the sorted quantity of interest* $B_n^{(i)}$
  ▷*Generate probability intervals; equation 2*
  for $n = 1 : N$ **do**
    $P_n^{(i)} = p_0^i \frac{N-n}{N}$
  end for
  ▷*CCDF: Concatenate vectors* $P_n^{(i)}$, $B_n^{(i)}$ *and sample* $\tilde{X}_n^{(i)}$
  $E_n = [P_n^{(i)}, B_n^{(i)}, \tilde{X}_n^{(i)}]$
  ▷*Begin lower levels of subset simulation*
  for $i = 1 : m - 1$ **do**
  ▷*Set threshold*
    $b_i = B_{N-N_c}^{(i-1)}$
    ▷*Set seeds using equation 6*
    for $j = 1 : N_c$ **do**
      $n = N - N_c + j$
      $s_j^{(i)} = \tilde{X}_n^{(i-1)}$
    end for
    ▷*Generate conditional samples using Metropolis Hastings algorithm*
    $[X_n^{(i)}, R_n^{(i)}] = \text{MH\_I}(s_j^{(i)}, N_s, C, b_i)$
    $B_n^{(i)} \leftarrow R_n^{(i)}$ *Sort distances in descending order*
    $\tilde{X}_n^{(i)} \leftarrow X_n^{(i)}$ *Reorder the input samples to correspond to the sorted quantity of interest* $B_n^{(i)}$
    ▷*Generate probability intervals; equation 2*
    for $n = 1 : N$ **do**
      $P_n^{(i)} = p_0^i \frac{N-n}{N}$
    end for
    ▷*CCDF: Discard all rows after* $E_{i(N-N_c)}$
    ▷*Concatenate* $P_n^{(i)}$, $B_n^{(i)}$, $\tilde{X}_n^{(i)}$ *and append to* E
    for $n = 1 : N$ **do**
      $E_{i(N-N_c+n)} = [P_n^{(i)}, B_n^{(i)}, \tilde{X}_n^{(i)}]$
    end for
  end for
  return E
end function

The simulation begins with level 0 Direct Monte Carlo where a set of $N = 100$ samples $\{X_n^{(0)} : n = 1, ..., 100\}$ are drawn from a Gaussian distribution centered at $O = [0, 0]^{\mathrm{T}}$ as shown in Fig. 5(a). The quantity of interest $\{R_n^{(0)} = H(X_n^{(0)}, C) : n = 1, ..., 100\}$ is the distance between each sample $X_n^{(0)}$ and the center of the circle $C = [3, -3]^{\mathrm{T}}$ (this is the equivalent of $Y_n^{(0)}$ used previously). This is determined by process $H(...)$ as defined by algorithm 1 in section II. If the condition $R_n^{(0)} \leq r_c^{(0)}$ is satisfied then the n^{th} sample $X_n^{(0)}$ is within the region F. This condition is used to determine if a sample is within the region F. The quantity of interest $R_n^{(0)}$ is sorted in descending order $\{B_n^{(0)} : n = 1, ..., 100\}$. This is because the samples with the lowest distances will be closest to the region F and have a higher likelihood of generating conditional samples closer to or within the region F than other samples as the simulation progresses to higher levels ($i > 0$). The input samples $X_n^{(0)}$ are reordered $\tilde{X}_n^{(0)}$ and correspond to the sorted quantity of interest $B_n^{(0)}$; to clarify, the distance between the sample $\tilde{X}_1^{(0)}$ and C is $B_1^{(0)}$. The probability intervals $P_n^{(0)}$ are determined by equation 2. The sorted quantity of interest $B_n^{(0)}$ and respective samples $\tilde{X}_n^{(0)}$ are concatenated with the probability intervals $P_n^{(0)}$ as shown in the column titled 'Level 0' in Fig. 6(a). The CCDF shown in Fig. 5(b) is generated by plotting the probability intervals $P_n^{(0)}$ against $B_n^{(0)}$. This CCDF shows that no samples have a distance less than the radius r_c therefore no samples have been drawn from the region F.

The SS method continues to the next level ($i = 1$) and generates N conditional samples using the MH method. The conditional samples $\{X_n^{(1)} : n = 1, ..., 100\}$ are generated from a set of seeds $s_j^{(1)} = \{\tilde{X}_{91}^{(0)}, ..., \tilde{X}_{100}^{(0)}\}$ that correspond to the sorted distances $\{B_n^{(0)} : n = 91, ..., 100\}$ from the previous level 0. The intermediate threshold $b_1 = B_{90}^{(0)}$ determined by equation 5 is used to ensure the conditional samples $X_n^{(1)}$ generated by each seed satisfies the condition $R_n^{(1)} \leq b_1$. The respective sample distances $R_n^{(1)}$ from C are less than or equal to the level 1 threshold b_1. This is to enable a progressive nature of drawing samples that are closer to the region F. The conditional samples are generated using algorithm 4. This will eventually lead to samples being drawn from the region F as SS proceeds to higher number of levels in the future. The level 1 threshold is marked by the dotted arc in Fig. 5(c). The figure shows chains of samples that lead to the region F. The distances $R_n^{(1)}$ of samples $X_n^{(1)}$ generated in level 1 are sorted in descending order $\{B_n^{(1)} : n = 1, ..., 100\}$. The input samples $X_n^{(1)}$ are reordered $\tilde{X}_n^{(1)}$ and correspond to the sorted distances $B_n^{(1)}$. The probability intervals $P_n^{(1)}$ are generated using equation 2 and concatenated with the sorted distances $B_n^{(1)}$ and their corresponding samples $\tilde{X}_n^{(1)}$. The table shown in Fig. 6(a) illustrates the conditional samples generated in level 1 using samples from level 0. The seeds used to generate samples in level 1 are discarded and replaced with the generated level 1 samples as illustrated in Fig. 6(b). Note the probability intervals $\{P_n^{(0)} : n = 91, ..., 100\}$, sorted distances $\{B_n^{(1)} : n = 91, ..., 100\}$ and the corresponding input samples $\{\tilde{X}_n^{(1)} : n = 91, ..., 100\}$ from level 0 that were used as seeds to generate the samples for level 1 are discarded and replaced with level 1 samples $\tilde{X}_n^{(1)}$ and their respective distances $B_n^{(1)}$ and probability intervals $P_n^{(1)}$. This process is repeated until the maximum number of levels m

Level 0 Level 1

$\mathbf{P}_n^{(0)}$	$\mathbf{B}_n^{(0)}$	$\tilde{\mathbf{X}}_n^{(0)}$	$\mathbf{P}_n^{(1)}$	$\mathbf{B}_n^{(1)}$	$\tilde{\mathbf{X}}_n^{(1)}$
$P_1^{(0)}$	$B_1^{(0)}$	$\tilde{X}_1^{(0)}$			
$\vdots$	$\vdots$	$\vdots$			
$P_{90}^{(0)}$	$B_{90}^{(0)}$	$\tilde{X}_{90}^{(0)}$			
$P_{91}^{(0)}$	$B_{91}^{(0)}$	$\tilde{X}_{91}^{(0)}$	$P_1^{(1)}$	$B_1^{(1)}$	$\tilde{X}_1^{(1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$P_{100}^{(0)}$	$B_{100}^{(0)}$	$\tilde{X}_{100}^{(0)}$	$P_{90}^{(1)}$	$B_{90}^{(1)}$	$\tilde{X}_{90}^{(1)}$
			$P_{91}^{(1)}$	$B_{91}^{(1)}$	$\tilde{X}_{91}^{(1)}$
			$\vdots$	$\vdots$	$\vdots$
			$P_{100}^{(1)}$	$B_{100}^{(1)}$	$\tilde{X}_{100}^{(1)}$

(a) Subset Simulation level 0 and level 1

$\mathbf{P}_n$	$\mathbf{B}_n$	$\tilde{\mathbf{X}}_n$
$P_1^{(0)}$	$B_1^{(0)}$	$\tilde{X}_1^{(0)}$
$\vdots$	$\vdots$	$\vdots$
$P_{90}^{(0)}$	$B_{90}^{(0)}$	$\tilde{X}_{90}^{(0)}$
$P_1^{(1)}$	$B_1^{(1)}$	$\tilde{X}_1^{(1)}$
$\vdots$	$\vdots$	$\vdots$
$P_{90}^{(1)}$	$B_{90}^{(1)}$	$\tilde{X}_{90}^{(1)}$
$P_{91}^{(1)}$	$B_{91}^{(1)}$	$\tilde{X}_{91}^{(1)}$
$\vdots$	$\vdots$	$\vdots$
$P_{100}^{(1)}$	$B_{100}^{(1)}$	$\tilde{X}_{100}^{(1)}$

(b) Subset Simulation level 0 and level 1 concatenation

Fig. 6. The columns $\mathbf{P}_n^{(0)}$, $\mathbf{B}_n^{(0)}$ and $\mathbf{X}_n^{(0)}$ represent the probability intervals, sorted quantity of interest and respective samples generated by DMC in level 0 of subset simulation. The samples $\tilde{X}_{N-N_c+1}^{(0)}$ to $\tilde{X}_N^{(0)}$ are used as seeds to generate conditional samples using MH in the next level of subset simulation. The seeds samples are replaced with the conditional samples generated in the next level of subset simulation.

is reached. This is when $i = m - 1$. Fig. 5(d) shows the overall CCDF at level 1. The overall CCDF is used to estimate the probability of drawing samples from the region F as approximately $P_F = 0.02$.

This example demonstrates the progressive nature of Subset Simulation when used to generate conditional samples to realize the rare 'tail' region of the pdf. This feature of SS results in the empirical observation that SS requires significantly less samples when compared to naive DMC to obtain estimates with the same accuracy. Subset Simulation is useful for generating samples that progress to the distribution of interest.

The next section applies the Subset Simulation method with modifications to estimate the probability of conflict between air traffic.

V. Application of Subset Simulation for Airborne Conflict detection

The estimation of the probability of conflict P_c between air traffic is a useful metric for Conflict Detection & Resolution (CD&R) methods. Such methods can be used in piloted aircraft but are useful for UAS where an automated method for CD&R will be required as part of a Sense-and-Avoid system [5].

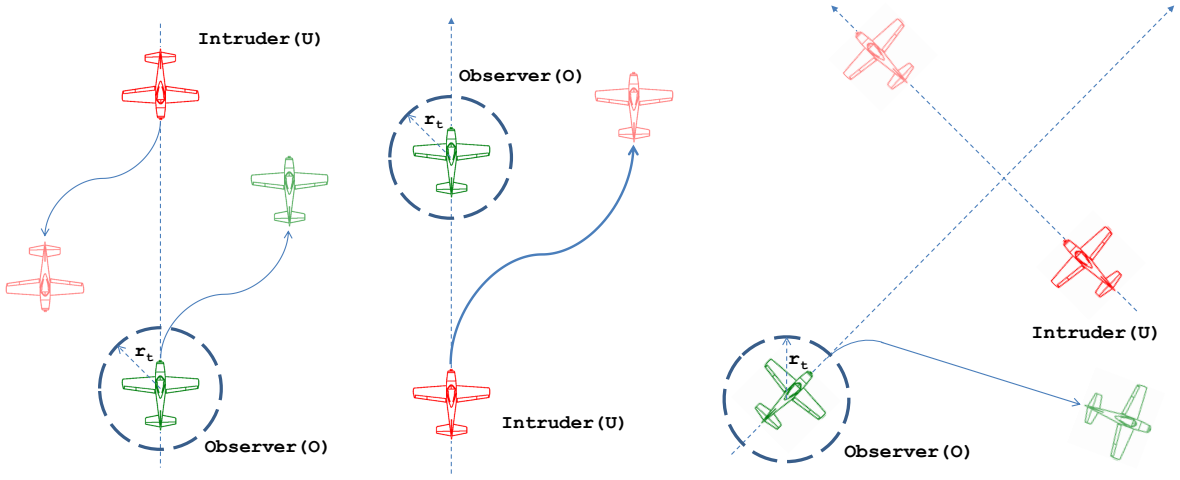
According to CAA CAP 393 Rules of the Air, the minimum lateral (Horizontal) separation required between two or more aircraft at any instance is 500 ft. A conflict event occurs when two or more aircraft collide or if there is a loss of this separation between them within a block of airspace. The conflict type depends on the geometry of the encounter between traffic, as defined in [13]. The 'Right-of-way' rules specified in 'Annex 2 to the Convention on International Civil Aviation – Rules of the Air' describe the conflict and the respective geometry of the encounter in a two-dimensional horizontal plane [10].

The conflicts and the respective encounters are illustrated in Fig. 7 as:

- A Head-on conflict scenario as shown in Fig. 7(a). In such a case each aircraft must turn right to avoid the collision.
- An Overtaking conflict scenario is where the aircraft being overtaken has the right of way as shown in Fig. 7(b). The overtaking aircraft must alter course right and keep clear of the overtaken aircraft. An overtaking condition exists while the overtaking aircraft is approaching the rear of another aircraft within an angle less than 70 degrees from the extended centreline of the aircraft being overtaken.
- A Converging conflict scenario is where the aircraft on the right has the right of way as shown in Fig. 7(c). The aircraft on the left must alter its course right to resolve the conflict.

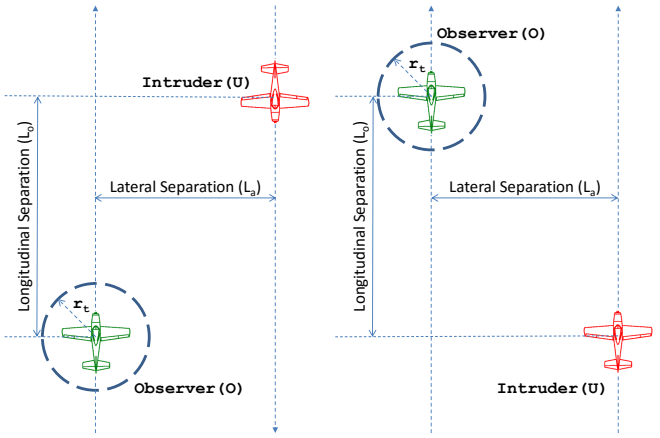
If a conflict is detected, the conflict type needs to be identified so that the appropriate resolution maneuver can be executed by the CD&R system to resolve the conflict. This paper addresses a key component of a detection of a conflict by estimating the probability of conflict P_c.

We assume a scenario where the traffic shares position information, however its intentions are unknown. The only information available regarding the state of traffic is its position. In such a scenario, CD&R system must allow for the possibility that the traffic is non-cooperative and may take inappropriate actions or may not adhere to the Rules of the Air. This type of situation requires a UAS to react and take appropriate action to ensure safe separation. To achieve this the P_c needs to be continuously evaluated against the behavior of the observed traffic so that the likelihood of the traffic



(a) Head-on (b) Overtaking (c) Converging

Fig. 7. These figures illustrate the geometric configuration of the different conflicts that might be encountered within a block of airspace. This includes different maneuvers required to be executed by the respective parities to resolve the conflict.



(a) Head-on pass (b) Intruder overtaking Observer

Fig. 8. The potentially conflicting scenarios based on the different conflicts shown in Fig. 7

causing a conflict can be calculated. Fig. 8 illustrates some potentially conflicting scenarios based on Fig. 7. During some phases of the scenario, the expected P_c can be very low; such as a magnitude of 10^{-8} (this is demonstrated later in this section). The previous sections have demonstrated that estimating low probabilities using the Direct Monte Carlo method is inefficient and this motivates the use of Subset Simulation (SS). Assessing the full pdf may not be feasible and may not be required. Subset Simulation provides an efficient method of determining the probability associated with all predicted conflicts thereby estimating P_c. In applying SS to this problem, P_c plays the role of the threshold of failure P_F.

The Subset Simulation method is used to estimate the probability of conflict P_c during the simulation of the potentially conflicting scenarios of the Observer and Intruder aircraft in the Head-on and Overtaking situations as shown in figures 8(a) and 8(b) respectively. Both scenarios show the Observer and Intruder in a non-conflicting state, where the Intruder is not within the Observer's protected zone. The Observer's protected zone is marked as a circle around the Observer with radius $r_t = 152.4$ m (500 ft). Although the current state is non-conflicting there is a potential for future conflict. For example from the Observer's perspective the Intruder could continue on its course or turn right or turn left. The latter could cause a loss of separation or worse – a collision between the Observer and the Intruder. Also in the situation when the lateral separation L_a between the Observer and Intruder is lower than or equal to the radius of the Observer's protected zone r_t; ($r_t \leq L_a$) a conflict occurs due to loss of separation or collision between the Observer and the Intruder. Therefore the likelihood of such conflict needs to be realized by estimating P_c.

The Subset Simulation method is used by the Observer to determine the probability of conflict P_c between itself and the approaching Intruder for the potentially conflicting scenarios shown in Fig. 8. However, since some parameters are not available this requires the method to be adapted. The order of magnitude for the target probability (conflict) region $(p_0)^m$ is unknown. The solution to this problem is addressed later in this section. Therefore the number of subset levels m required to reach the target probability level with a fixed p_0 is unknown. The Intruder and Observer are simulated as *nearly constant acceleration* point models [41]. This is a simple model that is used to illustrate the use of Subset Simulation. More complex dynamic models such as coordinated turn models [41] or Six-Degrees-of-Freedom (SixDoF) aircraft models [42] could be used instead. The simulation could be implemented in three-dimensional space, where the Observer's protected zone would be spherical or a cylindrical approximation with radius r_t as shown in [1]. The increase in complexity of the dynamic model or the number of dimensions will not affect the use of Subset Simulation and the computational advantages that it provides. This is evident in [35], where Subset Simulation was used to efficiently estimate the amount of propellant mass required by a spacecraft to perform attitude control. This involved the use of complex dynamic model in high dimensions.

In this paper, the dynamics of the Intruder and Observer are modeled in two-dimensional cartesian state space form as $U(K+1) = AU(K)$ and $O(K+1) = AO(K)$ respectively, where K is the time–step index. The Intruder and Observer state vectors are $U(K) = [x, u, a_x, y, v, a_y]^{\mathrm{T}}$ and $O(K) = [x, u, a_x, y, v, a_y]^{\mathrm{T}}$ respectively. The displacement, velocity and acceleration in the x-direction are represented by x, u and a_x respectively. The displacement, velocity and acceleration in the y direction are represented by y, v and a_y respectively. The state transition matrix A is defined as

$$A = \begin{bmatrix} 1 & \Delta T & \frac{1}{2}\Delta T^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta T & \frac{1}{2}\Delta T^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where ΔT is the period of discretized time-step. The sampling frequency $f = \frac{1}{\Delta T}$. The Observer estimates the state of the Intruder $\hat{U}(K)$ using a Kalman Filter [43], [44]. The periodic measurements of the Intruder's position $Z = [x, y]$ is defined in a cartesian coordinate frame. Alternatively, measurement model in the sensor coordinate frame could be could be considered [45]. This would not affect the application of Subset Simulation and the computational advantages that it provides. The cartesian space measurement equation used here.

$$Z = HU(K) + [w_x, w_y]' \tag{8}$$

where H is the measurement matrix.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{9}$$

$$w_x \sim \mathcal{N}(0, \sigma_x) \tag{10}$$

$$w_y \sim \mathcal{N}(0, \sigma_y) \tag{11}$$

The periodic position measurements are simulated by adding noise as w_x and w_y to the x and y directions respectively. The standard deviation of the measurement error in the x and y directions are σ_x and σ_y respectively. For the sake of simplicity the measurement noise is uncorrelated. The instantaneous state estimate of the Intruder is determined using a Kalman Filter as defined by algorithm 6, where $\hat{U}(K+1)$ and $\hat{S}(K+1)$ is the predicted state of the Intruder and the error covariance respectively.

The process noise covariance is Q. This is the *white-noise jerk* version of the *Wiener-Process Acceleration* model [41].

$$Q = \begin{bmatrix} Q_\sigma \frac{\sigma_{a_x}^2}{\Delta T} & 0 \\ 0 & Q_\sigma \frac{\sigma_{a_y}^2}{\Delta T} \end{bmatrix} \tag{12}$$

$$Q_\sigma = \begin{bmatrix} \frac{1}{20}\Delta T^5 & \frac{1}{8}\Delta T^4 & \frac{1}{6}\Delta T^3 \\ \frac{1}{8}\Delta T^4 & \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 \\ \frac{1}{6}\Delta T^3 & \frac{1}{2}\Delta T^2 & \Delta T \end{bmatrix} \tag{13}$$

where R is the measurement covariance.

Algorithm 6 Kalman Filter

function KF($\hat{U}(K)$, $\hat{S}(K)$, Z, H, Q, R, M_Z)
  ▷*Predict*
  $\hat{U}(K+1) = A\hat{U}(K)$
  $\hat{S}(K+1) = A\hat{S}(K)A^{\mathrm{T}} + Q$
  ▷*Update if new measurement is available*
  if $M_Z =$ true **then**
    $G = \hat{S}(K+1)H^{\mathrm{T}}\{[H\hat{S}(K+1)H^{\mathrm{T}}] + R\}^{-1}$
    $\hat{U}(K+1) = \hat{U}(K+1) + G\{Z - [H\hat{U}(K+1)]\}$
    $\hat{S}(K+1) = [I - GH]\hat{S}(K+1)$
  end if
  return $\hat{U}(K+1), \hat{S}(K+1)$
end function

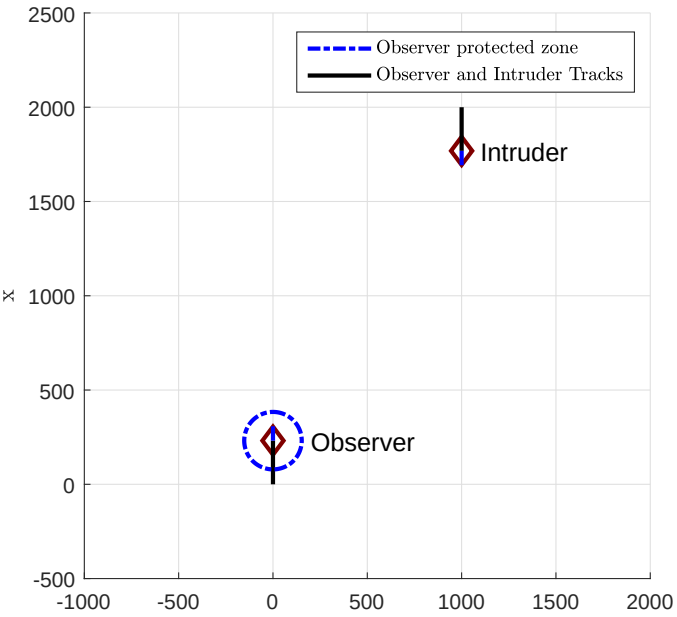


Fig. 9. Head-on pass scenario with 1000 m Lateral Separation

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{14}$$

and G is the Kalman gain.

A. Example

The Subset Simulation method is applied to the Head-on pass scenario, where the Intruder and Observer are both in cruise condition with lateral separation $L_a = 1000$ m and longitudinal separation $L_o = 2000$ m. The Observer's protected zone has a radius $r_t = 152.4$ m. The duration of the simulation $t = 20$ s with sampling frequency $f = 20$ Hz and the measurement frequency $f_M = 2$ Hz. The initial conditions of the Intruder and Observer are $U(0) = [2000\text{ m}, -77.2\text{ ms}^{-1}, 0, 1000\text{ m}, 0, 0]^{\mathrm{T}}$ and $O(0) = [0, 77.2\text{ ms}^{-1}, 0, 0, 0, 0]^{\mathrm{T}}$ respectively. The Intruder and Observer maintain constant velocity (the acceleration terms in the state vector remain zero) throughout the duration of the simulation. The nearly constant acceleration model is used instead of the constant velocity model to satisfy algorithm

requirements. The structure of the state vector of the former contains acceleration elements. The Intruder and Observer state vectors need to have acceleration elements to ensure they have the same dimensions as the state vectors of the samples generated by DMC and MH which have constant acceleration. In other words, the nearly constant acceleration model is used for propagating the Intruder and Observer (DMC and MH) samples to ensure equal dimensions. Since it is required to enable mathematical operations within the algorithms; such as, generating samples of the Intruder with constant acceleration and comparing the propagated trajectory with the Observer's trajectory.

Kalman Filter parameters:

- $\sigma_x = 0.1$ m
- $\sigma_y = 0.1$ m
- $\sigma_{a_x}^2 = 0.01$ m^{2}s^{-4}
- $\sigma_{a_y}^2 = 0.01$ m^{2}s^{-4}

Subset Simulation parameters:

- $N = 100$
- $p_0 = 0.1$
- $N_c = 10$
- $N_s = 10$
- $m = 7$

Ideally the SS method should continue to higher levels of simulation until conflicting samples are encountered and P_c can be estimated using the CCDF. This is assuming infinite simulation resources are available. This is impractical for implementation since simulation capacity is limited due to limited resources available. Therefore the SS method implemented requires a limited number of levels to be defined m.

Subset Simulation estimates $P_c(K+1)$ where $K+1$ is the time-step of an instance during the simulation as shown in Fig. 9. Subset Simulation begins with level 0 Direct Monte Carlo sampling. A set of 100 samples $\{U_n^{(0)} : n = 1, ..., 100\}$ representing the Intruder's pdf are drawn from the distribution that is centered at the Intruder's mean $\hat{U}(K+1)$ and covariance $\hat{S}(K+1)$. The mean and covariance are obtained from the Kalman filter defined in algorithm 6.

The set of samples $U_n^{(0)}$ and the intended vector of the Observer $O(K)$ are propagated to generate trajectories $J_{U_n}^{(0)}$ and J_O respectively. A trajectory J is a set of consecutive state vectors indexed by the time-step k. An initial state vector propagated for t seconds with sampling frequency f produces a trajectory consisting of tf state vectors indexed $k = 1, ..., tf$. For example the Observers state vector propagated for $t = 20$ s at sampling frequency $f = 20$ Hz produces the trajectory $J(O) = [O(1), ..., O(tf)] = [O(1), ..., O(400)]$, where $O(1)$ is the state vector of the Observer at time-step $k = 1$. The period the trajectory is propagated for is also the period of the simulation. The trajectories are propagated assuming that there is no subsequent maneuver – a constant acceleration is maintained during the propagation of the trajectory. Fig. 10(a) shows the Intruder samples and the respective trajectories generated with the projected position of the Observer during level 0 for a Head-on pass scenario with lateral separation $L_a = 1000$ m. No conflicting samples have been encountered yet. A conflicting sample is an Intruder sample $U_n^{(i)}$ generated

Algorithm 7 Propagate State to generate trajectory

function SAMPLETRAJECTORY(U_0, f, t, A)
  $J_0 = U_0$
  for $k = 0 : tf$ **do**
    $U(k+1) = AU(k)$
    $J(k+1) = U(k+1)$
  end for
  return J
end function

Algorithm 8 Determine miss-distance r and minimum points $\hat{U}_{xy}, O_{xy}$ between observer's trajectory J_O and Intruder trajectory $J_{\hat{U}}$

function MINDISTANCE(J_O, J_U)
  ▷*Difference between Observer and Intruder trajectory*
  $J_{OU} = J_U - J_O$
  ▷*Distance between each point on trajectories*
  $r_{OU} = \sqrt{J_{OU_x}^2 + J_{OU_y}^2}$
  ▷*Minimum distance*
  $r_{OU_{\min}} = \min(r_{OU})$
  ▷*Index of minimum distance*
  $k = \{r_{OU_n} | n = r_{OU_{\min}}\}$
  $J_{O_{\min}} = J_{O_{xy}}(k)$
  $J_{U_{\min}} = J_{U_{xy}}(k)$
  return $r_{O\hat{U}_{\min}}, J_{O_{\min}}, J_{\hat{U}_{\min}}$
end function

Algorithm 9 Estimating Probability of Conflict using Direct Monte Carlo

function PC_DMC($f, t, A, O, \hat{U}, \hat{S}, N, r_t$)
  $D = 0$
  ▷*Propagate Observer for t seconds*
  J_O = SAMPLETRAJECTORY(O, f, t, A)
  for $n = 1 : N$ **do**
  ▷*Draw sample*
    $U_n \sim \mathcal{N}(\hat{U}, \hat{S})$
    ▷*Propagate Intruder Samples for t seconds*
    J_n = SAMPLETRAJECTORY(U_n, f, t, A)
    ▷*Determine miss-distance between Observer and Sample Trajectories*
    r_n = MINDISTANCE(J_O, J_n)
    if $r_n \leq r_s$ **then**
      $D = D + 1$
    end if
  end for
  $P_c = \frac{D}{N}$
  return $P_c, D, U_n, J_O, J_n, r_n$
end function

in level i with a trajectory $J_n^{(i)}$ that has a miss-distance $r_n^{(i)}$ between the Observer trajectory J_O and satisfies the conflict condition $r_n^{(i)} \leq r_t$. The number of conflicting samples encountered in a level is D.

The quantities of interest are the miss-distances $\{r_n^{(0)} : n = 1, ..., 100\}$. These are the minimum distances between the Intruder samples' trajectories $\{J_{U_n}^{(0)} : n = 1, ..., 100\}$ and

the Observer trajectory J_O. Algorithm 8 defines the procedure to determine the miss-distances between the Observer and Intruder trajectories. A conflict is projected to occur when there is a loss of minimum separation between any sample in set J_{U_n} and the Observer trajectory J_O at any instance. The set of miss-distances $r_n^{(0)}$ are sorted in descending order $\{B_n^{(0)} : n = 1, ..., 100\}$. The input samples $U_n^{(0)}$ are reordered $\tilde{U}_n^{(0)}$ to correspond to the sorted miss-distances $B_n^{(0)}$. To clarify, the sample $\tilde{U}_1^{(0)}$ produces a trajectory $J_{\tilde{U}_1}$ that has the largest miss-distance $B_1^{(0)}$ between itself and the trajectory produced by the Observer J_O. The samples with lower miss-distances in the current level have a higher likelihood of generating conditional samples that satisfy the conflict condition than other samples in the current level. The vector of probability intervals $P_n^{(0)}$ are generated by

$$P_{n+1}^{(i)} = p_0^i \frac{N-n}{N} \quad n = 0, ..., (N-1) \tag{15}$$

Note the range of n in this equation is different to equation 2. This is due to the maximum number of levels limit m. In the event that SS reaches the maximum number of levels without encountering conflicting samples the probability of conflict will be estimated $P_c = P_N^{(m-1)} = P_{100}^{(m-1)} = 0$ (the last probability interval in the $P_n^{(m-1)}$ vector that is generated by equation 2) and this does not reflect the low magnitude of the probability. In contrast, the probability interval generated by equation 15 allows the probability of conflict to be estimated $P_c < P_{100}^{(m-1)}$; $P_{100}^{(m-1)} = 1 \times 10^{-8}$. This information means that although no conflicting samples have been encountered despite exhausting all levels of SS the expected P_c is estimated to be lower than $(p_0)^m$, the lowest probability level realizable due to the maximum number of levels limit reached by SS. Such information is more useful than the estimate $P_c = 0$ evaluated by equation 2. The level 0 CCDF is constructed by plotting the probabilities $P_n^{(0)}$ against $B_n^{(0)}$ as shown in Fig. 10(c). No conflicting samples have been drawn in level 0 since no miss-distances satisfy the conflict condition. If the number of conflicting samples $D > N_c$ then the probability of conflict is estimated $P_c = P_{(N-D+1)}^{(i)}$. This also applies for the situation where the maximum number of levels has been reached $i = m - 1$ and some conflicts have been encountered where the number of conflicts encountered is less than or equal to N_c; $(N_c \geq D > 0)$. The DMC method estimates the probability of conflict $P_c = \frac{D}{N}$ as defined by algorithm 9.

However if the condition $D > N_c$ is not satisfied and $i < m-1$; SS proceeds to the next level $(i > 0)$ and continues until the condition is satisfied or if the maximum number of levels is reached. This is because the conflict region of the pdf is not represented accurately enough due to the lack of sufficient samples representing the conflict region in the current level. Therefore it is necessary generate more conditional samples at higher levels of SS to progress towards representing the conflict region of the pdf more accurately.

The following subset levels $(i > 0)$ generate N conditional Intruder samples $\tilde{U}_n^{(i)}$ using the Metropolis Hastings method as defined in algorithm 10. The set of seeds $s_j^{(i)}$ required to generate the samples are selected from samples in the previous level using

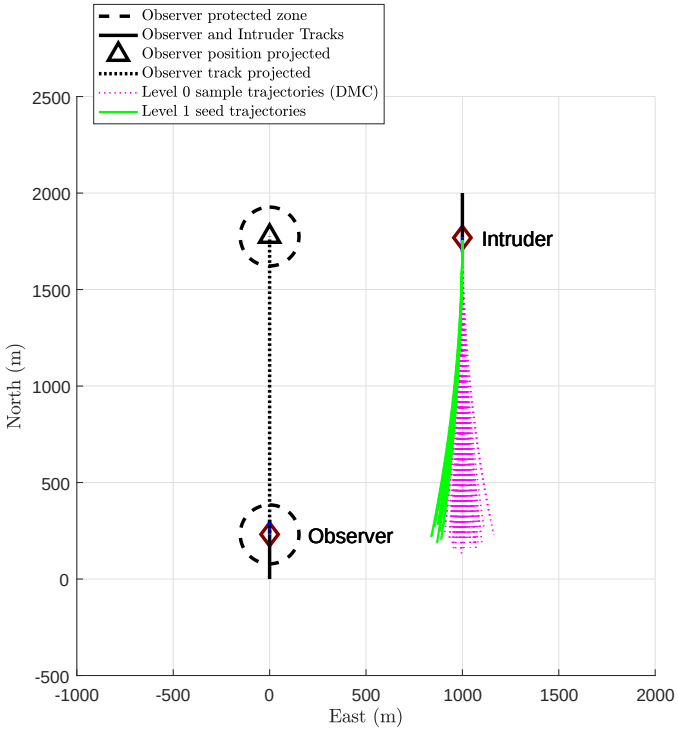
$$s_j^{(i)} = \tilde{U}_n^{(i-1)} \tag{16}$$

where $1 \leq j \leq N_c$, $(N - N_c + 1) \leq n \leq N$ and $i > 0$.

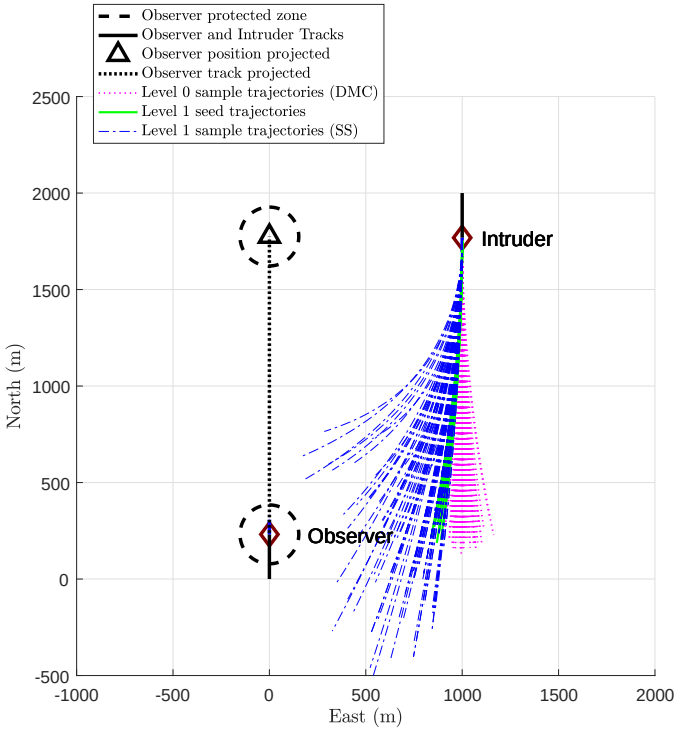
Fig. 10(a) highlights the trajectories of level 0 samples selected as seeds to generate level 1 conditional samples. Fig. 10(b) shows the trajectories of the conditional samples generated in level 1. The set $s_j^{(i)}$ contains N_c seeds; one for each chain. Each chain generates N_s samples. This maintains the total number of samples as N for each level. The MH

Algorithm 10 Generate conditional samples using Metropolis Hastings

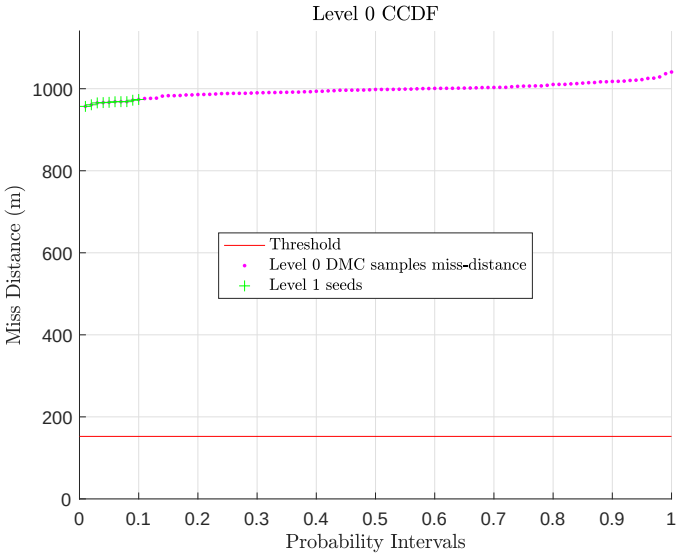
function MH_CONFLICTSAMPLES(f, t, A, O, $\hat{U}$, $\hat{S}$, s_j, N_s, r_t)
  $\sigma_{r_t}^2 = r_t^2 I_{2\times 2}$
  $J_O = \text{SAMPLETRAJECTORY}(O, f, t, A)$
  for $j = 1 : N_c$ **do**
    $U_0 = s_j$ ▷*Select seed sample*
    ▷*For each seed generate N_s samples*
    for $k = 0 : N_s - 1$ **do**
    ▷*Draw acceleration sample from mean*
      $a_x^* \sim \mathcal{N}(0, 1)$
      $a_y^* \sim \mathcal{N}(0, 1)$
      $g = [0, 0, a_x^*, 0, 0, a_y^*]^{\mathrm{T}}$
      ▷*Generate Candidate sample U^**
      $U^* = U_k + g$
      ▷*Propagate Samples for t seconds*
      $J_U^* = \text{SAMPLETRAJECTORY}(U^*, f, t, A)$
      $J_{U_k} = \text{SAMPLETRAJECTORY}(U_k, f, t, A)$
      ▷*Determine minimum miss-distance and (x, y) coordinates of minimum points between Observer and Sample Trajectories*
      $[r_k, J_{O_{\min}}, J_{U_{k_{\min}}}] = \text{MINDISTANCE}(J_O, J_{U_k})$
      $[r^*, J_{O_{\min}}^*, J_{U_{\min}}^*] = \text{MINDISTANCE}(J_O, J_U^*)$
      ▷*Indicator function for miss-distance*
      $d = \begin{cases} 1 & \text{if } r^* < r_t \\ 0 & \text{if } r^* \geq r_t \end{cases}$
      ▷*Calculate acceptance ratio*
      $\beta = \frac{p(J_{\hat{U}_{\min}}^* | J_{O_{\min}}^*, \sigma_{r_t}^2) q(U^* | \hat{U}, \hat{S})}{p(J_{U_{k_{\min}}} | J_{O_{\min}}, \sigma_{r_t}^2) q(U_k | \hat{U}, \hat{S})} d$
      $\alpha = \min\{1, \beta\}$
      $e \sim [0, 1]$
      ▷*Accept candidate sample, trajectory and miss-distance if $e < a$*
      $U_{k+1}^{(j)} = \begin{cases} U^* & \text{if } e < \alpha \\ U_k & \text{if } e \geq \alpha \end{cases}$
      $J_{k+1}^{(j)} = \begin{cases} J^* & \text{if } e < \alpha \\ J_k & \text{if } e \geq \alpha \end{cases}$
      $r_{k+1}^{(j)} = \begin{cases} r^* & \text{if } e < \alpha \\ r_k & \text{if } e \geq \alpha \end{cases}$
    end for
  end for
  return $U^{(j)}$, $J^{(j)}$, $r^{(j)}$
end function



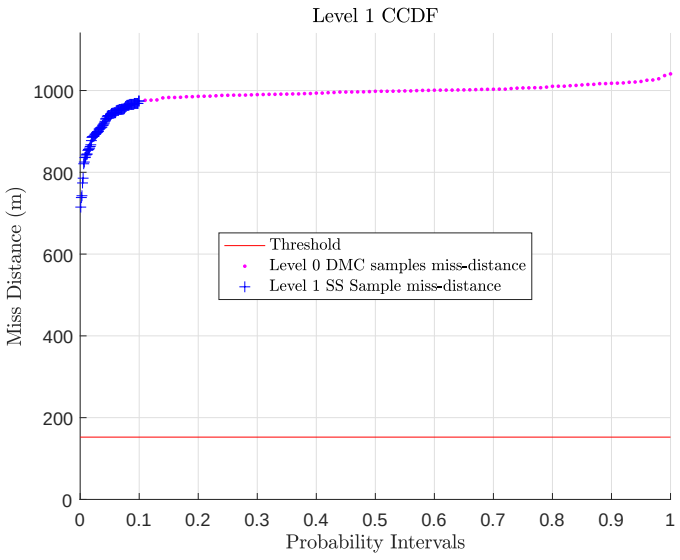
(a) Level 0 DMC sample trajectories with highlighted trajectories of samples



(b) Trajectories for Level 1 samples generated using N_c samples from level 0 used as seeds



(c) Level 0 CCDF with miss-distance of Level 1 seeds highlighted

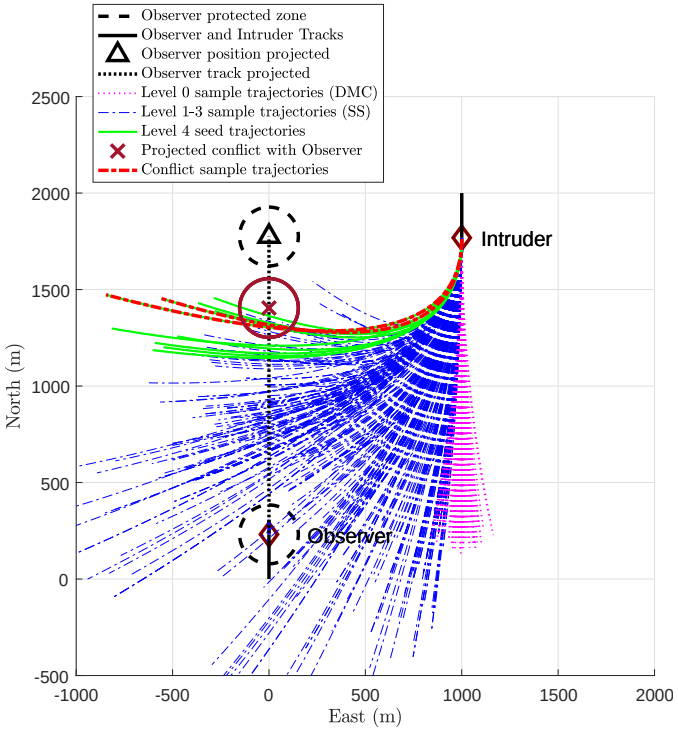


(d) Level 1 CCDF

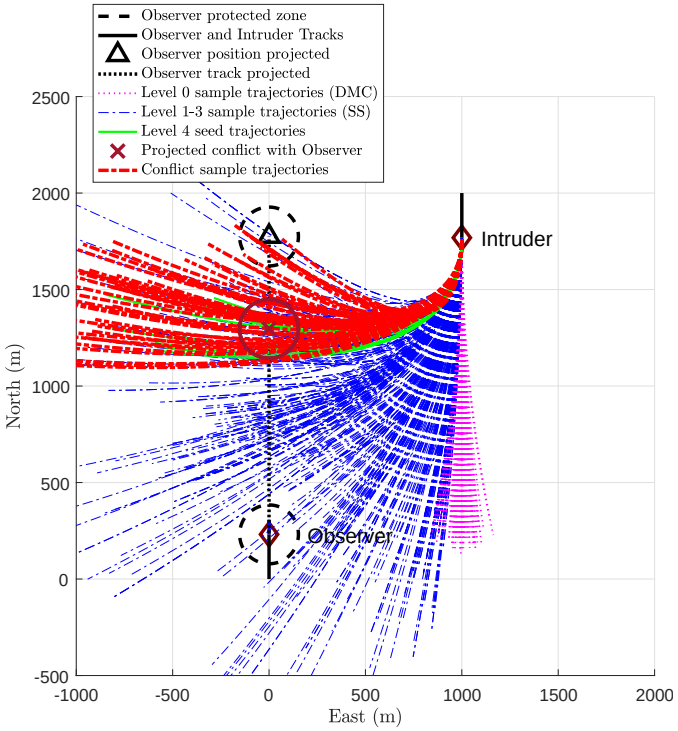
Fig. 10. These figures illustrate the application of SS to estimate the $P_c(K+1)$ at the time-step $K+1$ during a Head-on pass between an Observer $O(K)$ and Intruder $U(K)$ with lateral separation of 1000 m. SS begins with level 0 (DMC) where $N = 100$ samples are drawn from a distribution centered at the Intruder's state estimate $\hat{U}(K+1)$ with a covariance of $\hat{S}(K+1)$ obtained from the Kalman Filter. Fig. 10(a) shows trajectories generated by level 0 samples, no conflicting samples have been encountered. The simulation proceeds to level 1 where conditional samples are generated using N_c samples from level 0 as seeds. The trajectories of the level 0 samples used as seeds are highlighted in Fig. 10(a). The MH method is applied to generate conditional samples from the seeds. The trajectories of generated samples for level 1 are shown in Fig. 10(b). This process is continued to generates more trajectories as the number of levels increase. The method continues until conflicting samples are encountered at higher levels as shown in Fig. 11.

method uses an indicator d (as shown in algorithm 10) to ensure the miss-distance $r^{(i)*}$ between the Observer's trajectory J_O and Intruder trajectory $J^{(i)*}$ of the proposed sample $U^{(i)*}$ is less than the intermediate threshold b_i set by equation 5. If $r^{(i)*} > b_i$ then the proposed sample is rejected and the current sample of the Intruder is maintained.

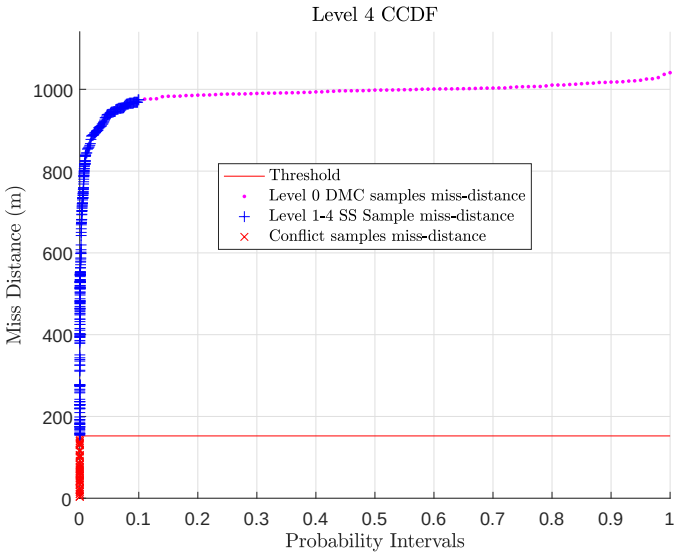
The miss-distances $\{r_n^{(1)} : n = 1, ..., 100\}$ of the conditional samples $U_n^{(1)}$ generated in level 1 are determined and sorted in descending order $B_n^{(1)}$ using the same method as level 0. The input samples $U_n^{(1)}$ are reordered $\tilde{U}_n^{(1)}$ to correspond to



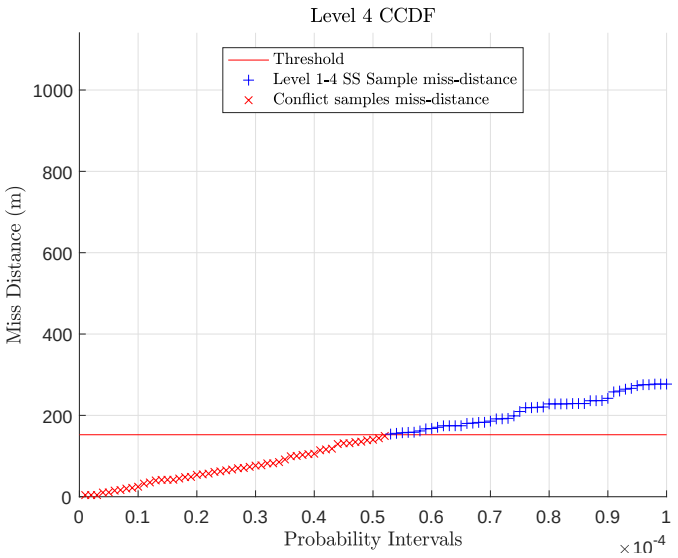
(a) Sample Trajectories for levels 0 to 3



(b) Samples Trajectories for levels 0 to 4



(c) Level 4 CCDF



(d) Level 4 CCDF

Fig. 11. The above figures show trajectories of conditional samples generated as the simulation continues to higher levels. Subset Simulation continues until the number of conflicting samples D found in a level is greater than N_c within a level as shown in Fig. 11(b). The probability of conflict is estimated as $P_c(K+1) = 0.52 \times 10^{-4}$ as shown in Fig. 11(d).

the sorted miss-distances $B_n^{(1)}$. The probability intervals $P_n^{(1)}$ for the current level are generated and plotted against $B_n^{(1)}$ to construct a CCDF. Fig. 10(d) shows the CCDF generated up to level 1. Note the miss-distances of the samples used as seeds from the previous level 0 (that are highlighted in Fig. 10(c)) are discarded and replaced with the miss-distances of the conditional samples generated in level 1. This illustrates that the samples used as seeds are discarded and replaced with the conditional samples generated in the current level. This process is repeated as SS progresses to higher levels until the condition $D > N_c$ is satisfied or the maximum number of levels is reached as defined in algorithm 11. Fig. 11(a) shows the trajectories of the conflicting samples encountered in level 3. However the condition $D > N_c$ had not been satisfied. This required SS to proceed to level 4 and generate conditional samples that satisfy the condition $D > N_c$ as shown in Fig. 11(b). The CCDF generated up to level 4 is shown in Fig. 11(c). The CCDF is used to estimate the

Algorithm 11 Estimate Probability of Conflict Using Subset Simulation

function PC_SS(f, t, A, O, $\hat{U}$, $\hat{S}$, N, r_t, p_0, m)
  $N_c = p_0 N$
  $N_s = p_0^{-1}$
  $i = 0$ ▷*Set current level*
  ▷*Direct Monte Carlo*
  $[D, U_n^{(i)}, r_n^{(i)}] =$ PC_DMC($f, t, A, O, \hat{U}, \hat{S}, N, r_t$)
  $B_n^{(i)} \leftarrow r_n^{(i)}$ *Sort distances in descending order*
  $\tilde{U}_n^{(i)} \leftarrow U_n^{(i)}$ *Reorder the input samples to correspond to the sorted quantity of interest* $B_n^{(i)}$
  ▷*Generate probability intervals; equation 15*
  for $n = 0 : N-1$ **do**
    $P_{n+1}^{(i)} = p_0^i \frac{N-n}{N}$
  end for
  ▷*CCDF: Concatenate vectors* $P_n^{(i)}$, $B_n^{(i)}$ *and samples* $\tilde{U}_n^{(i)}$
  $E_n = [P_n^{(i)}, B_n^{(i)}, \tilde{U}_n^{(i)}]$
  while $D < N_c$ **and** $i < m$ **do**
    $i = i + 1$
    $b_i = B_{N-N_c}^{(i-1)}$ ▷*Set threshold*
    ▷*Set seeds using equation 6*
    for $j = 1 : N_c$ **do**
      $n = N - N_c + j$
      $s_j^{(i)} = \tilde{U}_n^{(i-1)}$
    end for
    ▷*Metropolis Hastings to obtain conflicting samples*
    $[U_n^{(i)}, r_n^{(i)}] =$ MH_CONFLICTSAMPLES(f, t, A, O, $\hat{U}$, $\hat{S}$, s_j, N_s, b_i)
    $B_n^{(i)} \leftarrow r_n^{(i)}$ *Sort distances in descending order*
    $\tilde{U}_n^{(i)} \leftarrow U_n^{(i)}$ *Reorder the input samples to correspond to the sorted quantity of interest* $B_n^{(i)}$
    ▷*Generate probability intervals; equation 15*
    for $n = 0 : N-1$ **do**
      $P_{n+1}^{(i)} = p_0^i \frac{N-n}{N}$
    end for
    ▷*CCDF: Discard all rows after* $E_{i(N-N_c)}$
    ▷*Concatenate* $P_n^{(i)}$, $B_n^{(i)}$, $\tilde{U}_n^{(i)}$ *and append to* E
    for $n = 1 : N$ **do**
      $E_{i(N-N_c+n)} = [P_n^{(i)}, B_n^{(i)}, \tilde{U}_n^{(i)}]$
    end for
    $D = |B_n^{(i)} \leq r_t|$ ▷*Number of conflicts* D
  end while
  if $D > 0$ **then**
    $P_c = P_{(N-D+1)}^{(i)}$
  else
    $P_c = P_N^{(i)}$ ▷*No conflicting samples were found select lowest probability interval*
  end if
  return P_c, E
end function

Algorithm 12 Determine Probability of Conflict using SS and DMC

$O(0)$ ▷*Initialize Observer*
$U(0)$ ▷*Initialize Intruder*
$\hat{U}(0)$ ▷*Initialize Intruder Estimate*
$\hat{S}(0)$ ▷*Initialize Intruder Covariance*
$M_c = 0$ ▷*Measurement counter*
for $K = 0 : tf$ **do**
  $O(K+1) = AO(K)$ ▷*Propagate Observer*
  $U(K+1) = AU(K)$ ▷*Propagate Intruder*
  $M_Z = false$ ▷*Flag to indicate new measurement*
  if $M_c = \frac{f}{f_M}$ **then** ▷*Conduct Intruder position measurement*
    $Z = HU(K+1) + [w_x, w_y]^{\mathrm{T}}$
    $M_Z =$ true ▷*Set flag to indicate that new measurement is available for Kalman filter Update*
    $M_c = 0$ ▷ *Reset measurement counter*
  end if
  $M_c = M_c + 1$ ▷*Increment measurement counter*
  ▷*Predict/Update estimate of Intruder with Kalman filter*
  $[\hat{U}(K+1), \hat{S}(K+1)] =$ KF($\hat{U}(K), \hat{S}(K), Z, H, Q, R, M_Z$)
  ▷*Estimate Probability of Conflict using Subset Simulation*
  $P_c^{(\mathrm{SS})}(K+1) =$PC_SS($f$, t, A, O, $\hat{U}(K+1)$, $\hat{S}(K+1)$, N, r_t, p_0, m)
  ▷*Estimate Probability of Conflict using Direct Monte Carlo*
  $P_c^{(\mathrm{DMC})}(K+1) =$PC_DMC($f, t, A, O, \hat{U}(K+1), \hat{S}(K+1), N, r_t$)
end for

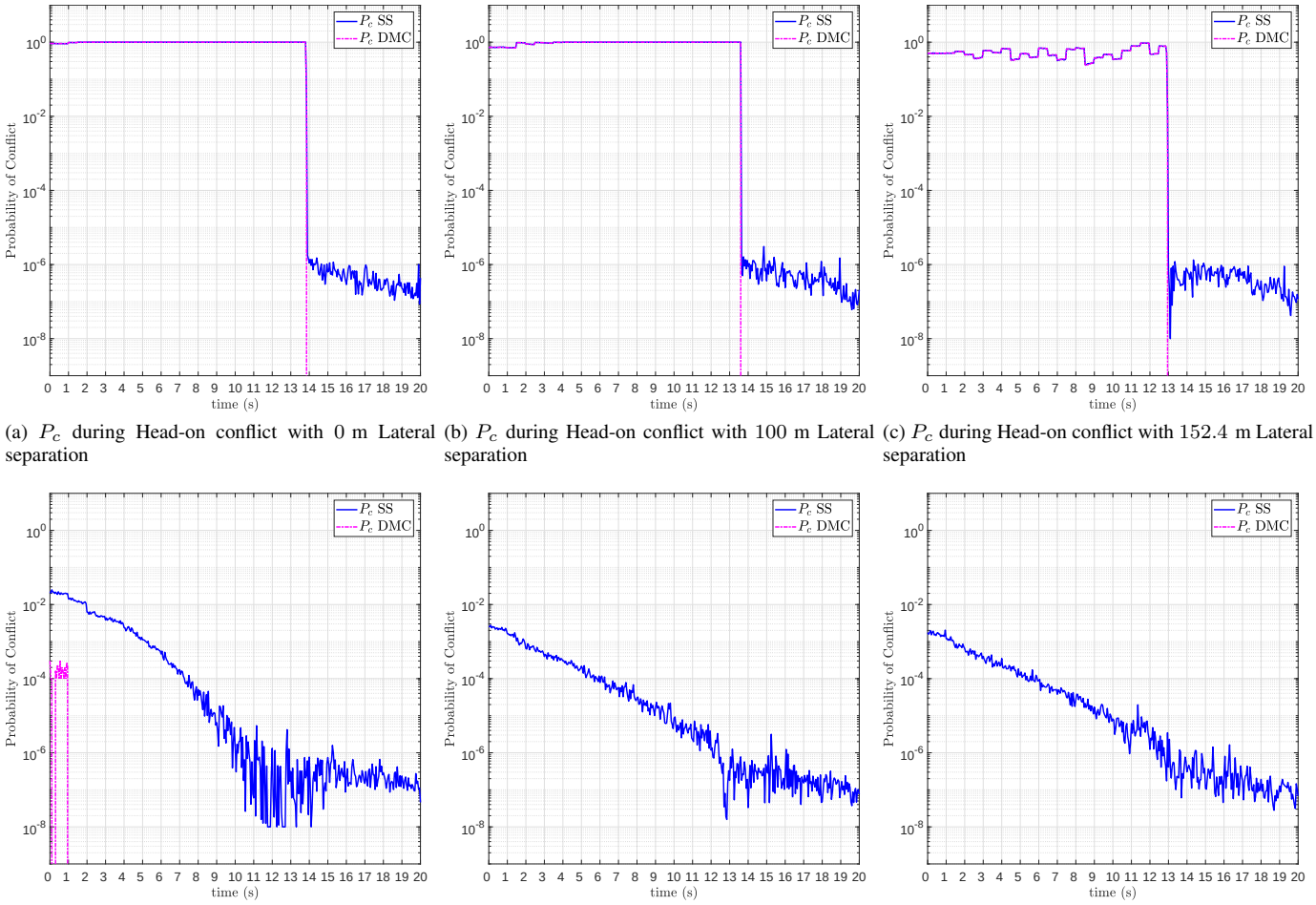
$P_c(K+1) = 0.52 \times 10^{-4}$ as shown in Fig. 11(d). This process is repeated throughout the duration of the simulation to determine the probability of conflict for each time-step using samples from the prediction of the Intruder's estimate $\hat{U}(K+1)$ and covariance $\hat{S}(K+1)$.

VI. RESULTS

The Subset Simulation method has been tested and compared with the Direct Monte Carlo (DMC) method to estimate the probability of conflict P_c between the Observer and Intruder by simulating the scenarios shown in Fig. 8. The Observer and Intruder were simulated with constant velocity in a geometric configuration based on the three different types of conflict shown in Fig. 7. The P_c metric was estimated as an average of 50 Monte Carlo simulations during the Head-on and Overtaking conflicts as shown in figures 8(a) and 8(b) respectively. The tests were repeated with varying lateral separations $L_a = \{0, 100, 152, 500, 1000, 1100\}$ m.

The following Subset Simulation parameters were used for all scenarios: $N = 100$; Level probability: $p_0 = 0.1$; $N_c = p_0 N = 10$; $N_s = \frac{1}{p_0} = 10$; $m = 7$; Observer minimum separation threshold $r_t = 500$ ft $= 152.4$ m. Algorithm 12 defines the simulation conducted.

The number of samples used for each level of SS remains constant. However the number of levels required at a given time-step varies depending on the magnitude of P_c. Therefore



(a) P_c during Head-on conflict with 0 m Lateral separation

(b) P_c during Head-on conflict with 100 m Lateral separation

(c) P_c during Head-on conflict with 152.4 m Lateral separation

(d) P_c during Head-on pass with 500 m Lateral separation

(e) P_c during Head-on pass with 1000 m Lateral separation

(f) P_c during Head-on pass with 1100 m Lateral separation

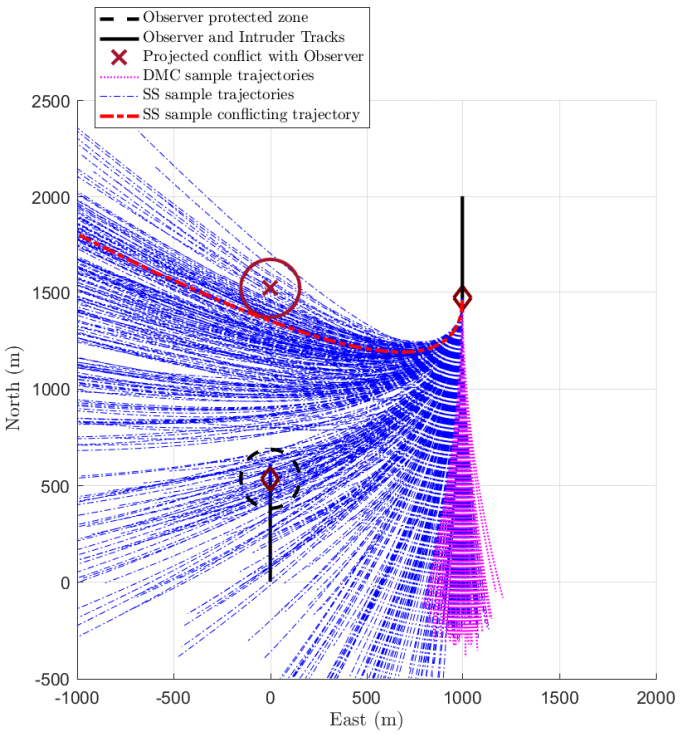
Fig. 12. The estimated P_c using the Subset Simulation and Direct Monte Carlo methods during the Head-on pass as shown in Fig. 8(a) with varying lateral separation $L_a = \{0, 100, 152, 500, 1000, 1100\}$ m.

the total number of samples N_T required to realize a conflict at a given time-step varies as a function of time-step. In the interest of a fair comparison of the computational effort between the two methods, an equal number of samples are evaluated to estimate the probability of conflict using both methods. The estimation using DMC is conducted with N_T samples, where N_T is the number of samples that are used in the SS method at the same time-step. To clarify, if the SS method reaches level $i = 4$ to satisfy the conflict condition for estimating the $P_c^{(\mathrm{SS})}(K)$ at time-step K, then $N_T = 100 \times 5 = 500$ samples have been used by the SS method. Therefore DMC estimates the $P_c^{(\mathrm{DMC})}(K)$ for the same time-step with 500 samples only. The approximate number of samples and the time taken to simulate each scenario is shown in table I in the appendix.

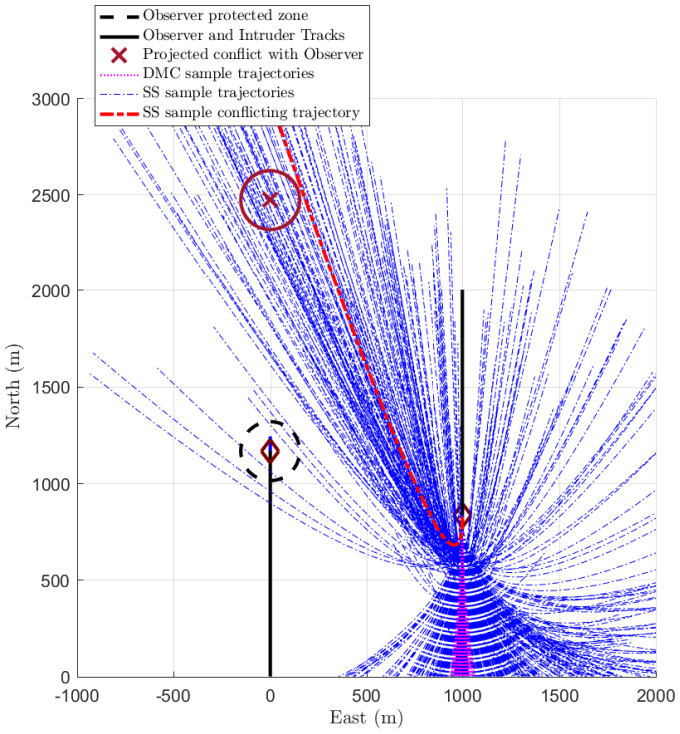
A. Estimation of P_c for Head-on Pass scenario

The Intruder and Observer parameters used for the Head-on pass scenario are as follows: The Intruder and Observer maintain a constant speed of 150 kt (77.17 ms^{-1}). The Observer maintains a constant heading of $0°$; the Intruder maintains a constant heading of $180°$. The Observer's minimum separation threshold is $r_t = 500$ ft $= 152.4$ m. The Longitudinal separation is $L_o = 2000$ m.

Figures 12(a), 12(b) and 12(c) show the estimation of P_c for the Head-on pass scenario using SS and DMC methods with lateral separations of 0 m, 100 m and 152 m respectively. The scenarios are conflicting because the geometric configuration and initial conditions of both the Observer and Intruder are conflicting and remain as such throughout the duration of the simulation. When $t \leq 12$ s the Intruder and Observer are approaching each other the estimated P_c increases. This is as expected because a conflict is imminent. Both estimation methods show approximately the same P_c as expected, since the first level of the SS method is DMC sampling. At this stage the conflict region of the pdf is large and the probability of drawing a sample which leads to a conflict is high. The conflict occurs at $t \approx 12.5$ s due to the loss of separation between the Observer and Intruder. Fig. 12(c) shows the estimation of P_c with lateral separation $L_a = r_t = 152.4$ m. This is a conflicting scenario since the Intruder skims Observer's protected boundary at $t \approx 12.5$ s as the Observer and Intruder pass each other. The oscillations during $t \leq 12$ s are due to $L_a = r_t$. This is a borderline situation.



(a) Head-on conflict scenario with 1000 m Lateral separation before head-on pass.



(b) Head-on conflict scenario with 1000 m Lateral separation after pass.

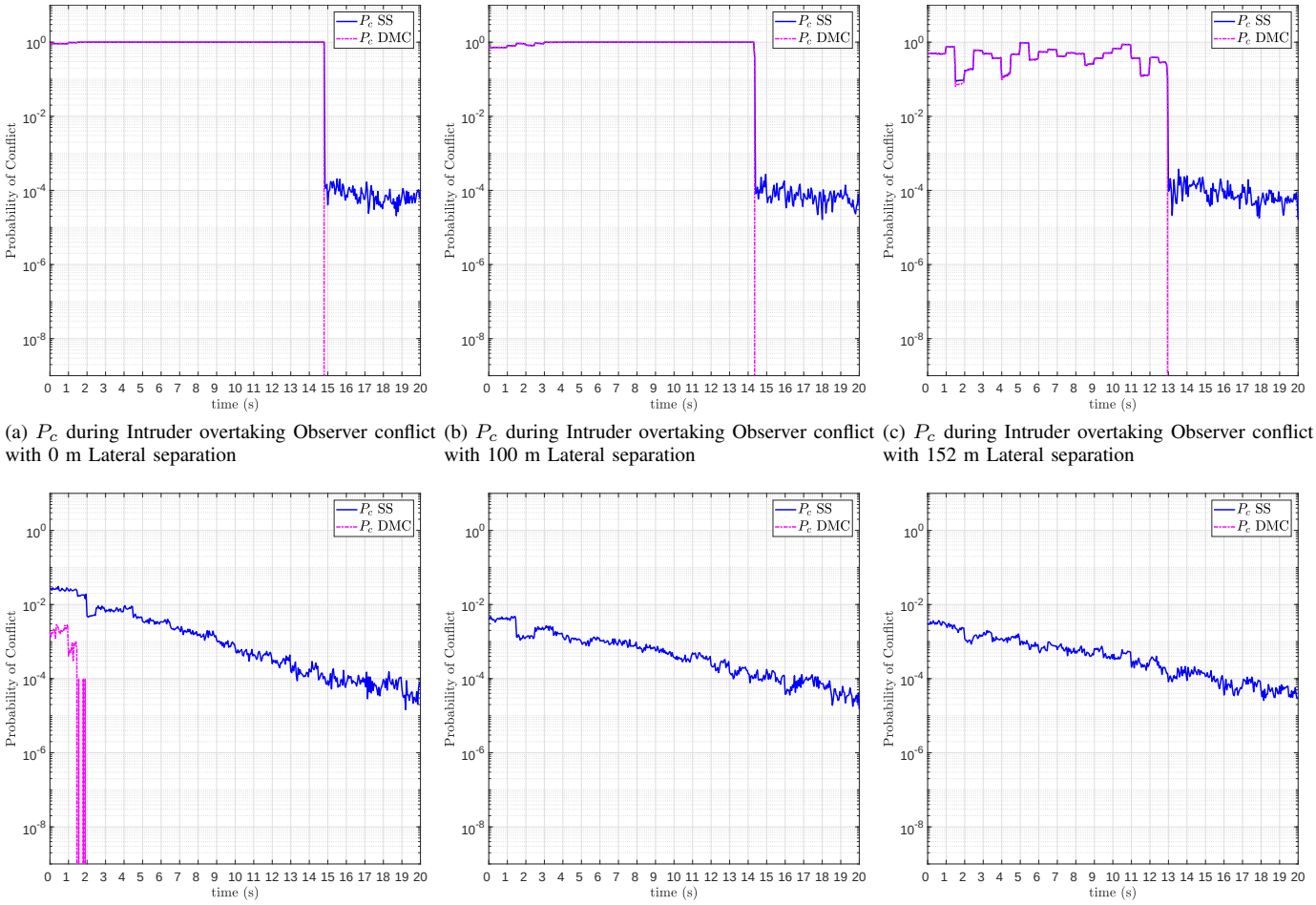
Fig. 13. SS and DMC trajectories for Head-on pass with lateral separation 1000 m.

The Intruder and Observer pass each other at $t \approx 13$ s. The P_c estimated by both methods is still 1 until $t > 14$ s where the Intruder has exited the Observer's protected zone. At this stage the Observer and Intruder have receding relative velocities and are moving away from each other. P_c is expected to reduce at this stage as shown in the log-y plot. The conflict region of the pdf reduces since both Intruder and Observer are moving away from each other. The SS method estimates the P_c as being close to zero at an order of magnitude of 10^{-7}. The lowest probability which can be realized is $P_c = 10^{-8}$. This is due to a maximum level restriction imposed in the simulation. In such instances the probability of conflict can be considered to be less than the order of 10^{-8}. At this stage the DMC method draws the same number of samples as SS but is unable to find conflicting samples and estimates $P_c = 0$. This is because the region of conflict within the pdf has reduced and the probability of drawing a conflicting sample is rare. This requires the DMC method to draw and evaluate a larger number of samples at this stage before a conflicting sample is drawn from the rare region of conflict within the pdf. The SS method is able to obtain the conflicting samples from the rare region of the pdf by generating samples conditionally in such a way that the samples satisfy the intermediate thresholds leading to the rare region using the MH method. Each subset level corresponds to an intermediate threshold. This progressive feature of the SS method allows a more efficient approach to reach the rare 'tail' region of the pdf.

As the lateral separation of the scenario is increased, the expected P_c decreases. The scenario is simulated with a lateral separation of 500 m, 1000 m and 1100 m as shown in figures 12(d), 12(e) and 12(f) respectively. These are non-conflicting scenarios. The figures show abrupt variations in P_c. These are caused by the Monte Carlo nature of our algorithm. Note that, since the sampling frequency is high relative to the thickness of the line in the figure, the variations in P_c are particularly readily perceived. The conflict region of the pdf is smaller than the previous scenarios. The SS estimation method is able to estimate low P_c throughout the duration of the simulation, whereas with an equivalent number of samples the DMC method is unable to find conflicting or near conflicting samples of the Intruder in most instances. Fig. 12(d) shows abrupt variations in the P_c estimated by the DMC method when $t < 1$ s where the estimate tends to zero. These are instances where the DMC method is unable for find any conflicting samples and estimates $P_c = 0$.

Figures 13(a) and 13(b) show the trajectories of the samples evaluated by SS and DMC methods at an instance before and after the Intruder and Observer pass each other respectively. The progressive nature of the SS method can be observed as a concentration of trajectories leading to the conflict trajectory. In contrast the DMC method has drawn the same number of samples (most are overlapping) without realizing any conflicts.

Note, the accelerations for some of the trajectories are high. This is because the acceleration of the samples drawn from the proposed distribution are not limited. Limiting the acceleration would limit the accuracy of the estimation, since low probabilities might not be estimated. The intruders accelerations are



(a) P_c during Intruder overtaking Observer conflict with 0 m Lateral separation

(b) P_c during Intruder overtaking Observer conflict with 100 m Lateral separation

(c) P_c during Intruder overtaking Observer conflict with 152 m Lateral separation

(d) P_c during Intruder overtaking Observer with 500 m Lateral separation

(e) P_c during Intruder overtaking Observer with 1000 m Lateral separation

(f) P_c during Intruder overtaking Observer with 1100 m Lateral separation

Fig. 14. The P_c is estimated using the SS and DMC methods during the Intruder Overtaking the Observer scenario as shown in Fig. 8(b) with varying lateral separation $L_a = \{0, 100, 152, 500, 1000, 1100\}$ m.

not limited.

B. Estimation of P_c for Intruder Overtaking Observer

The scenario parameters used are as follows: The Intruder speed is 300 kt $=$ 154.3 ms^{-1} and the Observer speed is 150 kt $=$ 77.17 ms^{-1}. Both Intruder and Observer maintain a constant heading of 180°. The longitudinal distance L_o between the Intruder and Observer is $L_o = 1000$ m.

Both SS and DMC methods have been applied to the Overtaking scenario as shown in Fig. 8(b). Similar to the previous scenario, the SS method is able to obtain samples from the rare conflicting region of the pdf consistently throughout the duration of the simulation for this scenario. As the lateral separation increases, the P_c decreases (as expected). Figures 14(e) and 14(f) show the P_c when the lateral separation is 1000 m and 1100 m respectively. The change in P_c is less abrupt compared to the 100 m lateral separation after the Intruder as passed the Observer when $t > 13$ s. The P_c is approximately the same throughout the duration of the simulation. This is because the increased lateral separation includes samples with low turn rates in the conflict category and these are common enough to be drawn by the DMC method and SS method. With low lateral separation the conflicting samples will need high turn rates. These are rare and are realized by using SS method. In contrast the DMC method is unable to realize them. Also throughout the simulation, the relative change in angle of the Intruder from the Observer's perspective reduces as the lateral separation is increased. The conflicting samples can have lower turn rates despite the Intruder having passed the Observer. Such samples are common and can be realized by both methods.

VII. Accuracy and Efficiency of Subset Simulation

A range of magnitudes of probabilities have been evaluated within the simulated scenarios shown in the previous section. This section analyzes the accuracy and efficiency of using the Subset Simulation and Direct Monte Carlo methods to estimate probabilities at each of a number of orders of magnitude. In order for a fair comparison to be conducted – a common phase within a simulation scenario must be found where both methods are able to realize conflicting samples and estimate the probability of conflict.

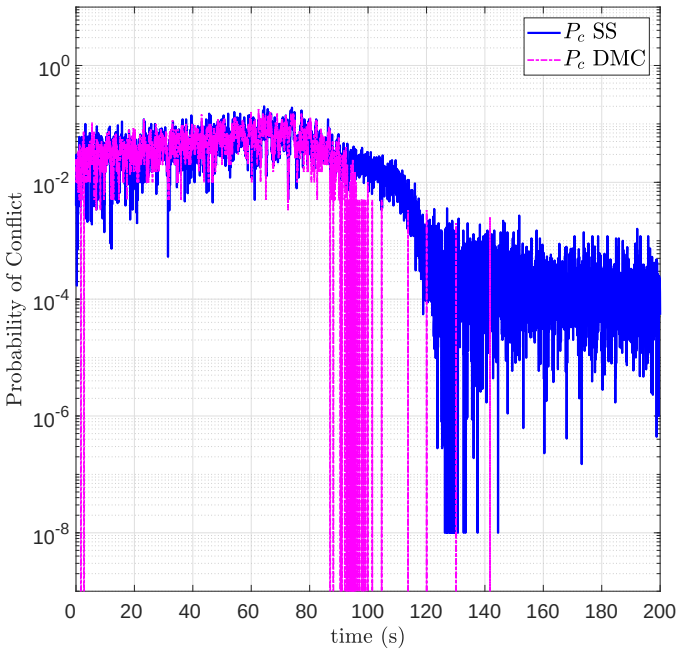


Fig. 15. Head-on pass scenario with 1000 m lateral separation and 20 km longitudinal separation

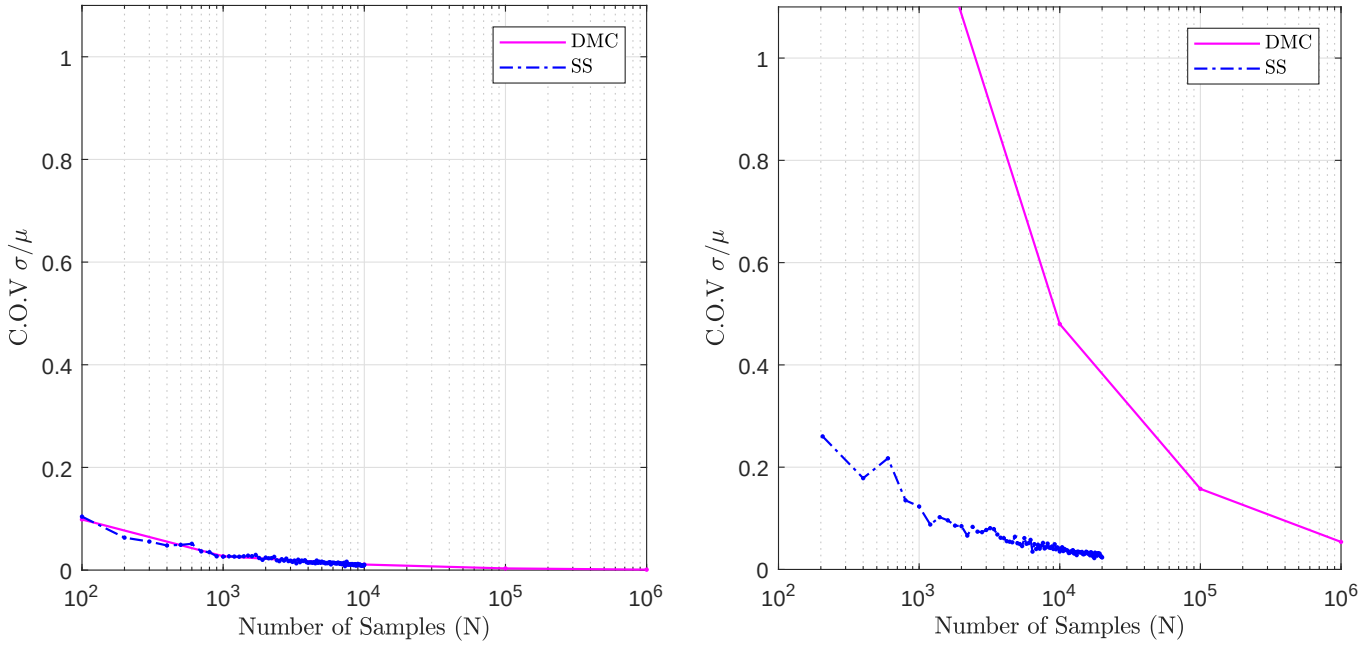
The first order of magnitude considered for comparison is $P_{c_1} \approx 10^{-1}$. A suitable phase to conduct the comparison is at $t = 1$ s during the Head-on scenario with lateral separation $L_a = 152.4$ m and longitudinal separation $L_o = 2000$ m where a conflict is inevitable. At this phase $p_0 \leq P_{c_1} < 1$ and both methods estimate a similar probability of conflict. This is as expected since the probability is large enough to generate sufficient conflicting samples in the first level of Subset Simulation and it does not progress to higher levels of Subset Simulation. The first level of Subset Simulation is Direct Monte Carlo so the performance is the same.

The second order of magnitude considered is P_{c_2}. This probability needs to be lower than P_{c_1} where $P_{c_2} < p_0$. Such phases occur frequently in the Head-on pass and Overtaking scenarios, typically when $t > 14$ s as shown in figures 12 and 14 respectively. Note, during such phases the Subset Simulation method is able to obtain conflicting samples and provide a good estimate for P_c. However, the Direct Monte Carlo method fails to find conflicting samples and is unable to estimate the probability of conflict accurately (other than in a trivial case, $P_c = 0$ that is inaccurate). For example the Head-on pass scenarios in Fig. 12 shows abrupt changes in P_c in some cases from a magnitude of 10^{-1} to 10^{-8} at approximately 13 s as the Observer and Intruder pass each other. This change in magnitude of probability is very large and abrupt (steep). The magnitude 10^{-8} is very rare. For such probabilities the Subset Simulation method is able to obtain conflicting samples and estimate the P_c but Direct Monte Carlo method fails to obtain conflicting samples and results in estimating $P_c = 0$. The Direct Monte Carlo method requires a large number of samples to estimate probabilities of such magnitude (10^{-8}). This might not be practical due to limited simulation resources. Therefore, this order of magnitude of probability is impractical for comparison since although the Subset Simulation method is able to find conflicting samples and estimate the P_c, the Direct Monte Carlo method is unable to find conflicting samples and fails to estimate the P_c.

In order to find a phase where P_{c_2} can be evaluated by both methods the simulation of the Head-on pass scenario with lateral separation of 1000 m was repeated once with increased longitudinal separation $L_o = 20000$ m for an increased period of $t = 200$ s. This allowed the change in P_c to occur less abruptly. Fig. 15 shows P_c estimated by Subset Simulation and Direct Monte Carlo methods during this scenario. Note, during the period $80 \text{ s} < t < 120$ s, there are frequent abrupt variations in the P_c estimated by the Direct Monte Carlo method as zero. These are phases where the method was unable to find a conflicting sample and estimated the probability of conflict as zero. A suitable phase for P_{c_2} is at $t = 100$ s where the probability of conflict estimated by Subset Simulation has reduced to approximately 10^{-2}; ($P_{c_2} \approx 10^{-2}$). This satisfies the $p_0 > P_{c_2}$ criteria. Also, it is the last phase after which the frequency of the Direct Monte Carlo method finding conflicting samples to estimate the P_c diminishes. In other words, it is the last phase where both methods are able to generate conflicting samples to estimate the probability of conflict for a comparison to be conducted.

The accuracy and efficiency are compared by calculating the coefficient of variance (c.o.v.) $\delta = \frac{\sigma}{\mu}$ for estimating the probabilities of conflict P_{c_1} and P_{c_2} using both Subset Simulation and Direct Monte Carlo methods for varying samples sizes N. The mean μ and standard deviation σ is calculated over 50 Monte Carlo runs. The sample intervals for Direct Monte Carlo are $N_{\text{dmc}} = \{10^2, 10^3, 10^4, 10^5, 10^6\}$ and the sample intervals for Subset Simulation are $N_{\text{SS}} = \{100n : n = 1, ..., 100\}$. Note, that N_{SS} is the number of samples at each level of Subset Simulation. The total number of levels can vary for each Monte Carlo run of Subset Simulation. This causes a total number of samples to vary for each Monte Carlo run. To allow a fair comparison an average of the total number of samples for each Monte Carlo run of Subset Simulation is used.

The c.o.v. for estimating P_{c_1} using Subset Simulation and Direct Monte Carlo methods at varying sample sizes N is shown by Fig. 16(a). Note both methods have similar c.o.v. as the average sample size increases. This is expected since the probability is large enough to be realized in level 0 of Subset Simulation that is Direct Monte Carlo. In Fig. 16(b) the c.o.v. of Subset Simulation for the lower probability of conflict P_{c_2} becomes significantly lower than the c.o.v. of DMC as the average number of samples is increased. A point of comparison between both methods can be made where the number of samples $N = 10^4$. Note that the c.o.v for Direct Monte Carlo is approximately 0.48 and the c.o.v for Subset Simulation is approximately 0.04. Also note that in order for the DMC method to achieve similar c.o.v as the Subset Simulation method it must use $N = 10^6$ samples. Therefore the Subset Simulation estimates probabilities of magnitude 10^{-2} approximately 10 times more accurately than the Direct Monte Carlo method while using a fraction of the samples (approximately $\frac{1}{100}$) that are required by the Direct Monte



(a) Coefficient of variance for varying number of samples using SS and DMC methods for estimating P_{c_1}

(b) Coefficient of variance for varying number of samples using SS and DMC methods for estimating P_{c_2}

Fig. 16. A comparison of accuracy and efficiency between DMC and SS methods for estimating the P_c during the Head-on pass scenario.

Carlo method to achieve similar levels of accuracy. The Direct Monte Carlo method requires at least 10^2 greater number of samples than the number of samples used by the Subset Simulation method to ensure a c.o.v that is similar to the c.o.v of the Subset Simulation method is achieved.

Assuming infinite simulation capacity is available, both methods could achieve the same level of accuracy by using infinite samples. However, this is impractical for implementation since finite simulation capacity is available due to limited resources. For such situations, Subset Simulation provides an improved accuracy compared to Direct Monte Carlo for estimating low probabilities.

VIII. Conclusion

This paper has demonstrated the utility of the Subset Simulation method to estimate the Probability of Conflict (P_c) between air traffic during conflicting and potentially conflicting scenarios based on the Rules of the Air defined by the International Civil Aviation Organization. These scenarios can be used to conduct benchmarks for comparing future algorithms. The Subset Simulation method has demonstrated the ability to seek samples from the rare conflict region of interest in an effort to estimate the probability of conflict with lower computational effort than Direct Monte Carlo method. For the equivalent number of samples, the Direct Monte Carlo method fails to consistently obtain samples from the region of interest within the probability distribution function.

This paper has also demonstrated the ability of Subset Simulation to estimate low probability of conflict (of magnitude 10^{-2}) approximately 10 times more accurately than the Direct Monte Carlo method while using approximately $\frac{1}{100}$ of the total samples used by the Direct Monte Carlo method to achieve the same level of accuracy as Subset Simulation. This has been demonstrated at a phase during a potentially conflicting scenario based on the Rules of the Air. This example situation has demonstrated that the Subset Simulation method is able to estimate low probabilities more accurately than Direct Monte Carlo method while using less samples than the Direct Monte Carlo method. We conclude that Subset Simulation method is more accurate and efficient than the Direct Monte Carlo method for estimating low probability of conflict between air traffic.

Apart from estimating the probability of conflict between air traffic in two-dimensions. The Subset Simulation method can be scaled for problems in higher dimension involving multiple intruders. For instance, within astrodynamics, space vehicles or satellites traveling from earth to geostationary orbit must avoid space debris (considered as intruders) encountered during its transition through low earth orbit. For such three-dimensional scenarios, the probability of conflict would need to be computed for each element of debris that is encountered. This would be useful for the resolution stage, where intruders can be prioritized based on the respective P_c and an optimized resolution maneuver determined to minimize the new P_c after the resolution maneuver.

The parameters used for Subset Simulation within this paper are typical values that have been used to estimate low failure probabilities in other disciplines [37]. The parameters might not be optimal for the problem explored in this paper. Future investigation will be focused on finding optimal parameters for estimating the probability of conflict between air traffic. A more efficient method of estimating the probability of conflict would be to modify the SS method further to use Sequential Monte Carlo Samplers instead of Markov Chain

TABLE I
THE NUMBER OF SAMPLES AND THE APPROXIMATE TIME TAKEN TO SIMULATE EACH SCENARIO.

Fig.	**Description**	**Number of samples** SS	**Number of samples** DMC	**Simulation time** (minutes)
12(a)	Head-on conflict between Observer and Intruder with 0 m lateral separation.	5,617,900	5,617,900	33
12(b)	Head-on conflict between Observer and Intruder with 100 m lateral separation.	5,745,700	5,745,700	35
12(c)	Head-on conflict between Observer and Intruder with 152.4 m lateral separation.	6,204,500	6,204,500	38
12(d)	Head-on pass between Observer and Intruder with 500 m lateral separation.	10,926,300	10,926,300	61
12(e)	Head-on pass between Observer and Intruder with 1000 m lateral separation.	11,514,300	11,514,300	64
12(f)	Head-on pass between Observer and Intruder with 1100 m lateral separation.	11,521,700	11,521,700	64
14(a)	Intruder overtaking Observer with 0 m lateral separation	4,223,700	4,223,700	23
14(b)	Intruder overtaking Observer with 100 m lateral separation.	4,448,500	4,448,500	24
14(c)	Intruder overtaking Observer with 152 m lateral separation.	5,193,400	5,193,400	28
14(d)	Intruder overtaking Observer with 500 m lateral separation.	8,058,900	8,058,900	44
14(e)	Intruder overtaking Observer with 1000 m lateral separation.	8,361,500	8361500	47
14(f)	Intruder overtaking Observer with 1100 m lateral separation.	8,449,800	8,449,800	48
16(a)	Estimated coefficient of variance for varying number of samples using SS and DMC methods for estimating P_{c_1}.	40,402,500	40,402,500	94
16(b)	Estimated coefficient of variance for varying number of samples using SS and DMC methods for estimating P_{c_2}.	53,027,650	53,027,650	1190

Monte Carlo [46]. This will allow the implementation to be parallelized in the seed selection stage and will give rise to improved statistical efficiency. We plan to investigate such improvements in future work.

ACKNOWLEDGMENTS

The authors would like to thank Matteo Fasiolo, Flávio De Melo, Elias Griffith and James Wright for their contributions. The authors would like to thank the reviewers for their constructive comments that improved the content and presentation of the paper. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Doctoral Training Grant.

APPENDIX

The simulations presented in this paper were executed on a computer with the following specifications:

- Processor: Intel© Core i7 - 3770k - @ 3.5 GHz
- Memory: 16.0 GB
- Operating system: Microsoft© Windows 7 64 bit
- Matlab© version: R 2015b - 4 worker threads

Table I shows the approximate time taken to execute each scenario with reference to the figures presented in this paper. For each scenario an equal number of samples are used to estimate the probability of conflict using both Subset Simulation and Direct Monte Carlo.

REFERENCES

[1] P. Angelov, *Sense and avoid in UAS: Research and Applications*. Chichester, West Sussex, United Kingdom: John Wiley & Sons, 2012.

[2] "Annex 13 to the Convention on International Civil Aviation - Aircraft Accident and Incident Investigation," International Civil Aviation Organization, 2013.

[3] H. P. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," DTIC Document, Stanford, Tech. Rep., 1980.

[4] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[5] J. K. Kuchar and L. C. Yang, "A Review of Conflict Detection and Resolution Modeling Methods," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 4, pp. 179–189, 2000.

[6] J. Krozel, M. E. Peters, and G. Hunter, "Conflict Detection and Resolution for Future Air Transportation Management," NASA, Tech. Rep. NASA CR-97-205 944, 1997.

[7] A. Warren, "Medium Term Conflict Detection for Free Routing: Operational Concepts and Requirements Analysis," in *16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings*, vol. 2, Oct 1997, pp. 9.3–27–9.3–34 vol.2.

[8] K. Zeghal, "A review of different approaches based on force fields for airborne conflict resolution," in *AIAA Guidance, navigation and control conference*, 1998, pp. 818–827.

[9] B. M. Albaker and N. A. Rahim, "A Survey of Collision Avoidance Approaches for Unmanned Aerial Vehicles," in *2009 International Conference for Technical Postgraduates (TECHPOS)*, Dec 2009, pp. 1–7.

[10] "Annex 2 to the Convention on International Civil Aviation - Rules of the Air," International Civil Aviation Organization, 2005.

[11] J. Kuchar and A. C. Drumm, "The Traffic Alert and Collision Avoidance System," *Lincoln Laboratory Journal*, vol. 16, no. 2, p. 277, 2007.

[12] "Federal Aviation Administration Part 91 - General Operating and Flight Rules," U.S Government Publishing Office, Federal Aviation Administration, 2015.

[13] "CAP 393 Air Navigation: The Order and the Regulations," Published by TSO (the Stationery Office) on behalf of the UK Civil Aviation Authority, Civil Aviation Authority, United Kingdom, 2015.

[14] "Annex 1 to 18 to the Convention on International Civil Aviation," International Civil Aviation Organization, 2012.

[15] C. Mishra, M. Mehta, E. J. Griffith, and J. F. Ralph, "Doing the Right Thing: Collision Avoidance for Autonomous Air Vehicles," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2013, pp. 2581–2586.

[16] C. Whitlock, "FAA records detail hundreds of close calls between airplanes and drones," Washington Post, 2015, [accessed 8 April 2016]. [Online]. Available: https://www.washingtonpost.com/world/national-security/faa-records-detail-hundreds-of-close-calls-between-airplanes-and-drones/2015/08/20/5ef812ae-4737-11e5-846d-02792f854297_story.html?utm_term=.e40ec9856c73

[17] A. Mcfadyen, L. Mejias, P. Corke, and C. Pradalier, "Aircraft Collision Avoidance using Spherical Visual Predictive Control and Single Point Features," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 50–56.

[18] S. Huh, S. Cho, Y. Jung, and D. H. Shim, "Vision-Based Sense-and-Avoid Framework for Unmanned Aerial Vehicles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 4, pp. 3427–3439, Oct 2015.

[19] D. Accardo, G. Fasano, L. Forlenza, A. Moccia, and A. Rispoli, "Flight Test of a Radar-Based Tracking System for UAS Sense and Avoid," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1139–1160, APRIL 2013.

[20] P. J. Nordlund and F. Gustafsson, "Probabilistic Noncooperative Near

Mid-Air Collision Avoidance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1265–1276, 2011.

[21] P.-J. Nordlund and F. Gustafsson, "Probabilistic Conflict Detection for Piecewise Straight Paths," Linköping Universitet, Tech. Rep., 2008, tech. Rep. 2871. [Online]. Available: http://www.control.isy.liu.se/publications/

[22] F. Lindsten, P.-J. Nordlund, and F. Gustafsson, "Conflict Detection Metrics for Aircraft Sense and Avoid systems," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*. Elsevier, 2009, pp. 65–70.

[23] H. A. Blom and G. Bakker, "Conflict Probability and Incrossing Probability in Air Traffic Management," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3. IEEE, 2002, pp. 2421–2426.

[24] V. P. Jilkov, X. R. Li, and J. H. Ledet, "Improved estimation of conflict probability for aircraft collision avoidance," in *17th International Conference on Information Fusion (FUSION) 2014*. IEEE, 2014, pp. 1–7.

[25] J. P. Chryssanthacopoulos and M. J. Kochenderfer, "Accounting for State Uncertainty in Collision Avoidance," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 951–960, 2011.

[26] T. B. Wolf and M. J. Kochenderfer, "Aircraft collision avoidance using Monte Carlo real-time belief space search," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 2, pp. 277–298, 2011.

[27] F. Belkhouche, "Modeling and Calculating the Collision Risk for Air Vehicles," *IEEE transactions on vehicular technology*, vol. 62, no. 5, pp. 2031–2041, 2013.

[28] H. Blom, J. Krystul, and G. Bakker, "Free Flight Collision Risk Estimation by Sequential Monte Carlo Simulation," in *Stochastic Hybrid Systems*, ser. Automation and Control Engineering Series, C. Cassandras and J. Lygeros, Eds. Taylor & Francis CRC Press, November 2006, vol. 24, pp. 247–279. [Online]. Available: http://doc.utwente.nl/66885/

[29] O. Watkins and J. Lygeros, "Stochastic Reachability for Discrete Time Systems: An Application to Aircraft Collision Avoidance," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 5, Dec 2003, pp. 5314–5319 Vol.5.

[30] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 4, pp. 199–220, 2000.

[31] J. Krozel and M. Peters, "Strategic Conflict Detection and Resolution for Free Flight," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 2, Dec 1997, pp. 1822–1828 vol.2.

[32] M. Prandini, H. A. Blom, and G. Bakker, "Air traffic complexity and the interacting particle system method: An integrated approach for collision risk estimation," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 2154–2159.

[33] S.-K. Au and J. L. Beck, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Engineering Mechanics*, vol. 16, no. 4, pp. 263–277, 2001.

[34] S. Au and J. Beck, "Subset Simulation and its Application to Seismic Risk Based on Dynamic Analysis," *Journal of Engineering Mechanics*, vol. 129, no. 8, pp. 901–917, 2003.

[35] D. P. Thunnissen, S.-K. Au, and E. R. Swenka, "Uncertainty Quantification in Conceptual Design via an Advanced Monte Carlo Method," *JACIC*, vol. 4, no. 7, pp. 902–917, 2007.

[36] G. Schuller and H. Pradlwarter, "Benchmark study on reliability estimation in higher dimensions of structural systems an overview," *Structural Safety*, vol. 29, no. 3, pp. 167 – 182, 2007, a Benchmark Study on Reliability in High Dimensions. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167473006000373

[37] S.-K. Au and Y. Wang, *Engineering Risk Assessment with Subset Simulation*. Singapore Pte. Ltd., 1 Fusionopolis Walk, 07-01 Solaris South Tower, Singapore 138628: John Wiley & Sons, 2014.

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[39] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[40] C. P. Robert and G. Casella, *The Metropolis—Hastings Algorithm*. New York, NY: Springer New York, 1999, pp. 231–283. [Online]. Available: http://dx.doi.org/10.1007/978-1-4757-3071-5_6

[41] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, Oct 2003.

[42] R. C. Nelson, *Flight Stability and Automatic Control*. New York: WCB/McGraw Hill, 1998, vol. 2.

[43] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation: Theory Algorithms and Software*. New York: John Wiley & Sons, July 2001.

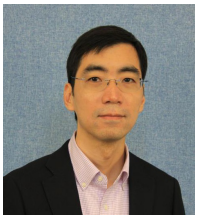
[44] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[45] X. R. Li and V. P. Jilkov, "A Survey of Maneuvering Target Tracking Part III : Measurement Models," *Signal and Data Processing of Small Targets, Proceedings of SPIE Conference on*, vol. 4473, no. August, pp. 1–24, 2001.

[46] P. Del Moral, A. Doucet, and A. Jasra, "Sequential monte carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, 2006.

Chinmaya Mishra received the B.Eng. degree and the Ph.D. degree in Electrical Engineering and Electronics from the University of Liverpool, Liverpool, U.K., in 2012 and 2016 respectively. He is a Research Associate in the Institute of Flight Systems Dynamics at the Technical University in Munich, Munich, Germany. His research interests are in guidance, navigation and safety systems in aerospace engineering, data fusion and Monte Carlo methods.

Simon Maskell Simon Maskell received the M.A., M.Eng. and Ph.D. degrees from the Department of Engineering, Cambridge University, Cambridge, U.K., in 1998, 1999, and 2003, respectively. Prior to 2013, Simon was a Technical Manager for Command, Control and Information Systems with QinetiQ, U.K.. Since 2013, he has been with the University of Liverpool, Liverpool, U.K., as a Professor of Autonomous Systems. His research interests include Bayesian inference applied to signal processing, multi-target tracking, data fusion, and decision support with particular emphasis on the application of Sequential Monte Carlo methods in challenging data science contexts.



Siu-Kui Au Siu-Kui Au holds a PhD (Caltech) in civil engineering and is the Chair of Uncertainty, Reliability & Risk in the Institute for Risk & Uncertainty, University of Liverpool. His research expertise includes engineering risk methods, Monte Carlo methods, Bayesian inference and field vibration testing.

Jason F. Ralph received a BSc degree (first class) in Physics with Mathematics from the University of Southampton in 1989 and a DPhil degree in Theoretical Physics from the University of Sussex in 1993. He is currently a Professor, and Former Head of Department, in Electrical Engineering and Electronics at the University of Liverpool, having been appointed as a Lecturer in 1999. His current research interests are in signal processing for quantum sensors, guidance and navigation systems in aerospace engineering, and data fusion.